Network coding for advanced video applications

Irina-Delia Nemoianu

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0034

EDITE – ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

Télécom ParisTech

Spécialité « Signal et Images »

présentée et soutenue publiquement par

Irina-Delia NEMOIANU

le 20 juin 2013

Codage Réseau pour des Applications Multimédias Avancées

Network Coding for Advanced Video Applications

Directeur de thèse : **Béatrice PESQUET-POPESCU**
Co-encadrement de la thèse : **Marc CASTELLA**

Jury

M. Corneliu BURILEANU, Professeur, Universitatea Politehnica Bucureşti, Bucarest	Président
M. Vladimir STANKOVIC, Professeur, University of Strathclyde, Glasgow	Rapporteur
M. Mihai CIUC, Professeur, Universitatea Politehnica Bucureşti, Bucarest	Rapporteur
M. Pascal FROSSARD, Professeur, École Polytechnique Fédérale de Lausanne, Lausanne	Examinateur
M. Marco CAGNAZZO, Maître de Conférences, Télécom ParisTech, Paris	Examinateur

T H È S E

Télécom ParisTech
Grande École de l'Institut Télécom – membre fondateur de ParisTech
46 rue Barrault — 75634 Paris Cedex 13 — Tél. +33 (0)1 45 81 77 77 — www.telecom-paristech.fr

Résumé

L'une des hypothèses fondamentales des réseaux classiques est que le transfert de données multi-saut de la source à la destination est géré au niveau des noeuds intermédiaires en transmettant les messages reçus par les liens en entrée à l'un ou plusieurs des liens en sortie, sans modifier le contenu du message. Si plusieurs flux de données partagent un noeud intermédiaire dans leur chemin, ce noeud intermédiaire va leur assigner de façon indépendante une priorité (ordonnancement) et un lien en sortie auquel les envoyer (routage).

Cette vision traditionnelle a récemment changé avec l'introduction du paradigme Codage Réseau (Network Coding, NC). Avec cette technique, chaque message transmis sur le lien en sortie d'un noeud est une fonction ou mélange des messages qui sont arrivés précédemment sur les liens d'entrée du noeud. Telle stratégie de mélanger, ou " encoder" les paquets au niveau des noeuds intermédiaires, ainsi que des moyens de décodage au récepteur, a été démontré surpasser le routage traditionnel en améliorant le débit du réseau et en réduisant au minimum le délai de livraison.

Le codage réseau a été appliqué à des nombreux types de communications ; en particulier, le scénario dans lequel NC a prouvé offrir le plus grand avantage est le multi-cast. Un résultat théorique très important stipule que, sous des hypothèses faibles sur le processus de mélange, en utilisant uniquement le réseau codage, il est possible d'atteindre un débit de multi-cast qui exploite pleinement la capacité du réseau.

Étant donné que ce résultat ne peut pas être atteint grâce au routage traditionnel seul, NC a été intégré avec succès dans une large gamme d'applications avec un grand avantage, comme a diffusion en temps-réel, le stockage distribué d'information, la livraison de contenu par des réseaux pair-à-pair (Peer-to-Peer, P2P), et communications interactives telles que les vidéo-conférences.

Le thème du codage réseau a reçu beaucoup d'attention de la communauté de recherche et a été approché à partir d'une multitude de disciplines, telles que la théorie des graphes, la théorie de l'information, la théorie du codage canal, la théorie de l'optimisation, *etc.*.

Dans cette thèse, nous nous concentrons sur l'application du paradigme du codage réseau pour des services de distribution de la vidéo en haute qualité. En particulier, nous étudions comment une technique de codage réseau peut être conçue et optimisée en profitant de la connaissance à la fois de la technique de codage vidéo utilisé pour produire le contenu et des propriétés débit-distorsion du contenu lui-même. À savoir, nous étudions l'intégration du codage réseau avec des techniques avancées de codage vidéo, tels que le codage par description multiple, qui est utilisé pour fournir une dégradation progressive du flux en présence de pertes, et le codage multi-vue codage, qui est utilisé pour fournir aux utilisateurs des nouveaux services vidéo interactifs 3D.

Nous étudions ces sujets dans différents scénarios de transmission, filaire comme non-filaire, de diffusion en temps-réel comme de services de mise en cache distribuée. Pour chacun de ces contextes, après un étude adéquat de l'état-de-l'art, nous fournissons nos propres contributions originales, qui ont fait l'objet de plusieurs publications internationales et que nous validons par une analyse de nos résultats expérimentaux.

Chapitre 1 — Codage réseau

Dans le premier chapitre, nous introduisons les concepts fondamentaux de codage réseau et nous présentons ses résultats théoriques les plus importants. De plus, nous donnons un aperçu de l'état-de-l'art, en termes des protocoles les plus efficaces pour le stockage et la transmission de données en utilisant le codage réseau.

Nous fournissons une analyse détaillée du codage réseau dans le contexte de la transmission vidéo, en discutant en détail les exigences et les contraintes spécifiques de la distribution vidéo . En outre, nous concentrons notre attention sur l'application du paradigme du codage réseau à des réseaux sans fil, ce qui présentente un défi supplémentaire à cause d'une capacité limitée et d'un taux de perte élevé.

Le concept du NC est apparu pour la première fois dans l'article de Ahlswede *et al.* [ACLY00], introduit par le désormais célèbre exemple du Réseau Papillon, présenté en Figure 1.

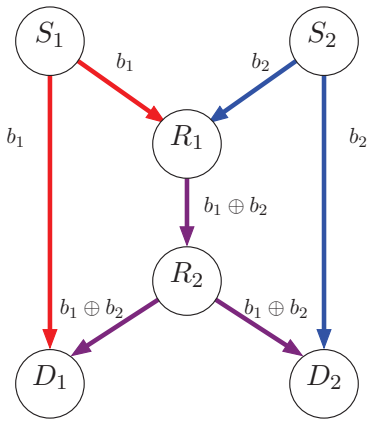


Figure 1: Le Réseau Papillon. Les arcs représentent des liens orientés avec une capacité de 1 message par seconde. Les noeuds source S_1 et S_2 veulent transmettre les messages b_1 et b_2 à deux noeuds déstination, D_1 et D_2. En envoyant le XOR de deux messages, $b_1 \oplus b_2$, cela peut être réalisé avec une seule transmission par lien.

Le problème du multi-cast dans un réseau filaire est ici considéré, avec deux sources S_1 et S_2 qui veulent livrer leurs messages b_1 et b_2 à deux destinations (sink), respectivement noeuds D_1 et D_2. Tous les liens sont supposés avoir une capacité d'un message par seconde. Si les noeuds intermédiaires R_1 et R_2 font seulement suivre les messages reçus, à chaque occasion d'envoi, ils peuvent soit livrer les deux messages à D_1, mais seulement b_1 à D_2, soit les deux messages à D_2 mais seulement b_2 à D_1. Autrement dit, le lien entre R_1 et R_2 devient un goulot d'étranglement.

Cependant, si le noeud R_1 peut envoyer une combinaison de b_1 et b_2 –par exemple, le OR exclusif bit-à-bit, ou XOR– comme représenté en Figure 1, les deux déstinataires peuvent obtenir à la fois les deux messages. Le sink D_1 peut décoder b_2 par XOR de $b1 \oplus b2$ avec le message b_1, reçu précédemment sur le lien direct avec S_1 ; D_2 peut récupérer b_1 de la même façon.

Le codage réseau peut donc obtenir un débit de multi-cast de deux messages par seconde, ce qui est mieux de ce que l'approche de routage peut atteindre, c'est à dire, au mieux 1.5 messages par seconde.

De manière plus générale, il a été montré [ACLY00] que dans le cas d' un réseau de communication point-à-point dans lequel une information est envoyée d'une source à un ensemble de destinations, si l'on utilise le codage réseau, la capacité de multi-cast est égale à la capacité de *min-cut* du réseau, ce qui n'est pas possible si l'on ne utilise que des

schémas de routage traditionnelles.

Stratégies de codage comme ces là présenté précédemment pour le réseau papillon impliquent qu'il y ait une certaine connaissance de la topologie du réseau et que des noeuds dédiés sont chargé d'effectuer les mêmes opérations de codage. Le schéma est donc centralisé et fixe.

Cependant, dans les réseaux réels, la structure, la topologie et les exigences de traffic peuvent changer rapidement et radicalement, tandis que les informations sur ces changements se propage avec un certain délai.

Dans les réseaux filaires, les capacités des liens peuvent varier en fonction de l'évolution des conditions de traffic et de congestion. Dans les réseaux d'overlay pair-à-pair, nombreux noeuds peuvent rejoindre ou quitter le réseau dans un court intervalle. Un réseau sans fil peut varier dans le temps en raison de l'attenuation des canaux, des interférences et de la mobilité des noeuds. Dans les réseaux sans fil ad-hoc, où les noeuds sont auto-organisés, les noeuds participants ont aussi des ressources limitées en termes de débit et de calcul, et la qualité de la transmission peut varier en raison de la mobilité des noeuds.

Dans des réseaux avec cycles et retards, les codes réseau peuvent être obtenues d'une manière répartie, en effectuant un codage réseau linéaire aléatoire (Random Linear Network Coding, RLNC) [HMS$^+$03], c'est à dire les coefficients des fonctions de codage linéaire sont choisis de façon indépendante et aléatoire dans un corps de Galois.

Dans le cadre de référence, le réseau est modélisé par un graphe acyclique $G(V, E)$ qui comporte un ensemble de noeuds V, un ensemble de liens E de capacité unitaire, une source $s \in V$ et un ensemble de déstinataires $T \subset V$.

La capacité de multi-cast h est la capacité du plus petit *cut* (*min-cut*) entre la source et chacun des déstinataires. Les h messages de la source sont indiquées par $x_1, \ldots, x_h$, et sont des symboles definis dans un corps de Galois $\mathbb{F}_q$, où q est l'ordre (taille) du corps.

La probabilité qu'un ensemble de coefficients choisi de manière aléatoire dans un corps de Galois $\mathbb{F}_q$ ne puisse pas assurer la décodabilité au récepteur est donné par : $p_e = 1 - \left(1 - \frac{|T|}{q}\right)^{|E|}$.

Ce résultat montre que la probabilité d'échec dans le cas des codes réseau aléatoires ne dépend pas du débit maximale de multi-cast h. De plus, en travaillant avec une taille de

corps q assez grand, la probabilité d'échec peut être rendue négligeable. Le problème reste alors dans la transmission des opérations de codage qui ont eu lieu sur chaque paquet.

Une approche pratique a été présenté par Chou *et al.* [CWJ03]. Depuis lors, celle là est devenue, sous la dénomination de codage réseau pratique (Practical Network Coding, PNC) - la norme *de facto* pour le codage réseau linéaire aléatoire, et a été appliquée avec succès pour résoudre plusieurs problèmes des applications multimédias. Nous présentons brièvement dans ce qui suit.

Toute noeud intermédiaire $v \in V$ a un ensemble liens intermédiaire $\Gamma_{in} = \{e'|out(e') = v\}$ et un ensemble de liens en sortie $\Gamma_{out} = \{e|in(e) = v\}$, comme représenté en Figure 2. Chaque lien $e \in E$ sortant du noeud v porte un symbole $y(e)$, qui est calculé comme une combinaison linéaire des symboles $y(e'_i)$ sur les liens d'entrée e' du noeud v, c'est à dire, $y(e) = \sum_{e' \in \Gamma_{in}} m_e(e') y_e(e')$ avec $m_e(e') \in F_q$. Le vecteur des coefficient $\mathbf{m(e)} = [m_e(e')]$ représente le vecteur de codage local pour le lien e.

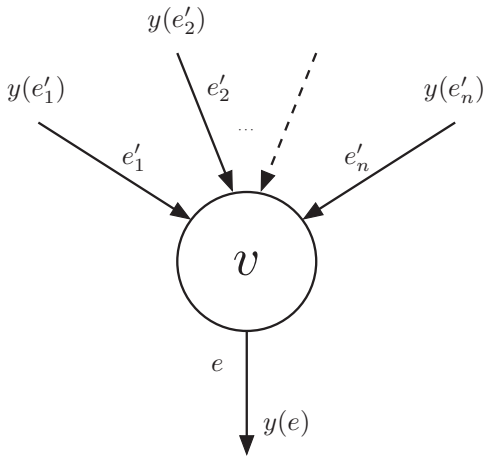


Figure 2: Noeud intermédiaire dans un réseau avec capacité de codage. $y(e'_i)$ représentent les symboles portés par les liens entrants e'_i dans le noeud v, tandis que $y(e)$ est le symbole transmis sur le lien e sortant par le noeud après la fonction de codage.

Par induction, le symbole $y(e)$ sur un lien peut être calculé comme une combinaison linéaire des symboles de source, c'est à dire, $y(e) = \sum_{i=1}^{h} g_i(e) x_i$. Les coefficients forment un vecteur de codage global $\mathbf{g}(e) = [g_1(e), \ldots, g_h(e)]$, qui est mis à jour à chaque opération de codage en utilisant les vecteurs de codage locaux $m(e)$. Le vecteur de codage global $\mathbf{g}(e)$ représente le symbole $y(e)$ du code en termes de symboles de source $x_1, \ldots, x_h$.

Lorsqu'un noeud sink $t \in T$ reçoit les symboles $y(e_1), \ldots, y(e_h)$, ils peuvent être

exprimés en termes des symboles de source comme :

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

où G_t est la matrice de codage global et la ligne i-ème de G_t est le vecteur de codage global correspondant à $y(e_i)$. Le sink t peut récupérer les h symboles de source tant que G_t est inversible, c'est à dire, est de rang plein h.

Cela est vrai avec une probabilité élevée dans la mesure où les coefficients de la matrice G_t sont choisis uniformément au hasard dans un corps de taille q suffisamment grande [HMS+03]. Les résultats expérimentaux [CWJ03] montrent que même des tailles de corps petites (corps du Galois $\mathbb{F}_{2^8}$ ou $\mathbb{F}_{2^16}$) fonctionnent bien dans la pratique et la probabilité d'échec devient négligeable.

Pour généraliser, chaque paquet dans le réseau peut être considéré comme un vecteur de symboles. Les relations algébriques ci-dessus restent valables pour les paquets si les symboles de la source sont regroupés en paquets. Afin d'être en mesure d'inverser le code à n'importe quel récepteur, les vecteurs d'encodage globaux doivent être obtenus à partir des paquets qui arrivent eux-mêmes. Cela se fait en faisant précéder le i-ème paquet source $\mathbf{x}_i$ avec le i-ème vecteur unitaire.

$$\begin{bmatrix} \mathbf{y}(e_1) \\ \vdots \\ \mathbf{y}(e_h) \end{bmatrix} = G_t \begin{bmatrix} 1 & \cdots & 0 & \mathbf{x}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & \mathbf{x}_h \end{bmatrix}$$

L'avantage de ces balises est que les vecteurs globaux de codage peuvent se trouvent dans les paquets eux-mêmes. Les sinks peuvent calculer G_t sans connaître la topologie du réseau ou les chemins de transfert de paquets. Le réseau peut donc être dynamique, avec des noeuds et des liens étant ajoutés ou supprimés de manière ad hoc.

En pratique, le flux de données est divisé en générations avec chaque génération comprenant h paquets de données. Les noeuds intermédiaires ne mélangent que des paquets

venant de la même génération. La source génère h nouveaux paquets de données en faisant précéder le i-ème vecteur unitaire à h dimensions à l'i-ème paquet de données. Les noeuds intermédiaires grandent les paquets reçus dans des buffer triés par numéro de génération et, quand il y a une occasion de transmission, le paquet qui doit être envoyé est généré par codage aléatoire de tous les paquets de la génération actuelle. Les paquets originaux sont obtenus au niveau des récepteurs en effectuant l'élimination de Gauss dès que h vecteurs linéairement indépendants ont été reçus.

Etant donné que, au but de décoder, un noeud récepteur doit attendre un nombre de paquets indépendants égale à la taille de la génération, ce paramètre affecte clairement le retard dans le réseau, de sorte qu'il est souhaitable de le maintenir aussi faible que possible. De plus, la taille de la génération donne la taille de l'overhead dû à la transmission des coefficients dans l'en-tête des paquets, et il doit aussi être maintenu petit. A titre d'exemple, un paquet IP sur Internet a une taille maximale de 1500 octets. Si chaque octet est considéré comme un symbole sur $\mathbb{F}_{2^8}$ et la taille de la génération est $h = 50$, alors l'overhead est d'environ $\frac{50}{1500} = 3.3\,\%$.

Cet aspect pourrait devenir important pour des communications sans fil où les paquets sont plus petits et l'overhead devient important. D'autre part, la taille de la génération affecte la façon dont les paquets sont " mélangés". La matrice de décodage serait de rang plein avec une probabilité d'au moins $\left(1 - \frac{1}{q}\right)^h$, il est donc souhaitable de disposer d' une taille de génération assez grande.

Chapitre 2 — Codage réseau pour la video en description multiple

Dans le deuxième chapitre, nous discutons la notion de codage à description multiple (*Multiple Description Coding*, MDC), une technique de codage destinée à fournir un compromis entre l'efficacité de l'encodage, la robustesse vis-à-vis des des pertes, et la complexité de calcul. Nous présentons les principes de base du MDC, et comment il peut être utilisé conjointement au codage réseau pour fournir à un flux vidéo, transmis sur un réseau peu fiable et sans retransmissions, tel que un réseau sans fil ad-hoc, robustesse vis-à-vis des erreurs et des pertes.

MDC est une technique de codage conjoint source-canal dans laquelle un flux de signal

unique est représenté par d sous-flux dénommé *descriptions* qui peuvent être livrés sur différents chemins [Goy01]. À la destination, les descriptions peuvent être décodées et reproduites de manière indépendante, et la qualité de la reconstruction s'améliore avec le nombre de descriptions reçues .

MDC est pensé pour fournir robustesse aux erreurs au flux encodé, en permettant n'importe quel sous-ensemble des descriptions d'être utilisé pour reconstruire le signal d'origine. De cette façon, des problèmes communs comme la congestion du réseau et la perte de paquets vont toujours provoquer une perte de la qualité, mais avec une dégradation progressive de la reconstruction plutôt qu'une interruption .

En plus de la robustesse aux l'erreurs, MDC permet également d'adapter le débit, comme les clients peuvent souscrire à une ou plusieurs des descriptions en fonction de leur capacité de téléchargement. Alors que dans la technique scalable, autre technique d'encodage largement utilisée, le flux multimédia est divisé en couches avec une couche de base et plusieurs couches d'amélioration, chacune dépendente des précédentes dans un ordre hiérarchique, en MDC toute description améliorer la qualité de la réconstruction, quel que soit l'ordre dans lequel elles sont reçues.

Pour la première contribution, nous avons formulé le problème de la diffusion d'un flux vidéo encodé dans plusieurs descriptions sur un réseau ad-hoc en termes de recherche d'un ensemble optimal de coefficients de combinaison. Ensuite, nous avons introduit une fonction objectif qui prend en compte l'effet que le décodage d'un nombre donné de descriptions a sur la distorsion totale. Ce cadre a été intégré avec un protocole inter-couche récemment proposé qui fournit à la fois une réseau de overlay acyclique et de la connaissance de l'état des voisins. Enfin, nous avons comparé les performances de notre technique avec la bien connue technique du codage réseau pratique combinée avec du flooding probabiliste.

Nous avons observé que la limitation de la taille de la génération au nombre de descriptions imposé par les contraintes sur le retard de la vidéo en temps réel, affecte gravement les performance de la technique de référence, ce qui en conséquence est constamment dépassé par l'approche proposée.

Cette technique , avec ses résultats expérimentaux, a été présenté à la Conférence Internationale ICASSP 2012.

Pour la deuxième technique , nous avons étendu l'utilisation du codage réseau avec fenêtre d'encodage variable (Expanding Window Network Coding, EWNC) au cas de la vidéo à descriptions multiples, afin de garantir la décodabilité instantanée

EWNC a été initialement proposé [VS10, VS12] pour fournir une protection inégale contre les erreurs (Unequal Error Protection, UEP) aux stratégies de codage aléatoire linéaire, pour la la transmission de messages contenant des paquets d'importance différente sur des liens avec perte.

L'idée principale de EWNC utilisée dans notre approche est d'augmenter la taille de la fenêtre d'encodage (c'est à dire, l'ensemble des paquets dans la génération qui peuvent apparaître en combinaison) pour chaque nouveau paquet. Si les vecteurs d'encodage sont conservés dans la mémoire tampon des récepteurs en forme échelonnée, en utilisant l'élimination de Gauss, cette méthode fournit la décodabilité instantanée des paquets si aucun pertes se produisent .

Cependant, l'efficacité de EWNC dépend fortement de l'ordre dans lequel les paquets sont inclus dans la fenêtre d'encodage. La méthode de EWNC originale a été proposée pour le codage vidéo en couches, donc la priorité des paquets était naturellement imposée par les dépendances entre les couches .

Une telle stratégie est irréalisable dans notre scénario : comme nous traitons avec plusieurs expéditeurs non coordonnés qui partagent un canal de diffusion, s'ils choisissent tous le même ordre de paquets (par exemple, ce là imposé par la structure en couches), à chaque occasion d'envoi ils allaient envoyer des combinaisons *non innovantes*.

Le scheduler optimisé débit-distorsion que nous proposons détermine l'ordre dans lequel les trames sont incluses dans la fenêtre d'encodage. Afin de réduire la probabilité de générer des paquets non-innovants, les sources opèrent une classification des trames (clustering) qui leur fournit un degré de liberté dans le choix des scheduling.

Nous avons comparé les performances de notre technique à EWNC appliquée à la fois sur des flux encodés à description simple et à description multiple, en supposant un scheduling trivial, et (dans le cas de MDC) de limiter les combinaisons aux trames de la même description.

Nous avons observé que l'introduction du scheduling, conjointement avec la possibilité

de mélanger des paquets de descriptions différentes, améliore de façon significative les performances par rapport aux techniques de référence, en termes de qualité vidéo perçue par l'utilisateur.

Cette contribution a été présentée lors de la 20ème Conférence Européenne de Traitement du Signal (EUSIPCO 2012), organisée par l'Association Européenne pour le Traitement du Signal, de la Parole et de l'Image (EURASIP).

Chapitre 3 — Codage réseau pour vidéo en contenu multi- vue

Dans le troisième chapitre, nous discutons de la notion de vidéo multi-vue (*Multi-View Video*, MVV), un paradigme vidéo émergent qui introduit de nouveaux services interactifs tels que la la télévision à point de vue libre et les téléconférences immersives.

MVV consiste dans la représentation simultanée d'une scène vue par N caméras placées dans différentes positions spatiales, appelés points de vue.

L'idée principale de notre contributions est d'utiliser le codage réseau à fenêtre variable afin de garantir la décodabilité instantanée du flux. Les trames sont incluses dans la fenêtre d'encodage dans un ordre déterminé par un scheduler optimisé débit-distorsion. Afin de réduire la probabilité de générer des paquets non-innovants, les sources utilisent un modèle probabiliste simplifié débit-distorsion qui leur donne un degré de liberté dans le choix du scheduling.

Nous avons comparé les performances de notre technique au codage réseau pratique appliqué à chaque vue indépendamment, et à la transmission sans codage réseau, les deux sous l'hypothèse d'un scheduling trivial.

Nous observons que l'introduction du scheduling, de concert avec la possibilité de mélange des paquets de vues différentes, améliore de manière significative la performance par rapport aux techniques de référence, en termes de qualité vidéo perçue par l'utilisateur.

Les résultats que nous avons obtenus suggèrent qu'une recherche dans tel sens pourrait être prometteuse, en particulier dans la direction d'une planification conjointe d'un protocole de gestion de l'overlay qui pourraient sélectionner quels noeuds du réseau devrait transmettre le flux.

Ces résultats ont été présentés à la conférence internationale VCIP 2012.

Chapitre 4 — Cache sociale repartie utilisant le codage réseau

Dans le quatrième chapitre, toujours dans le contexte de la vidéo multi-vue, plutôt que se concentrer sur la transmission en temps réel comme dans le chapitre précédent, nous nous concentrons sur un service de mise en cache distribuée qui, en raison du coût élevé en termes de débit de la transmission d'un flux multi-vue, peut fournir le grand avantage de soulager le serveur et le réseau d'une partie de leur charge.

En particulier, notre accent est mis sur la mise en cache coopérative de contenu vidéo multi-vue, un scénario dans lequel les utilisateurs qui ont récemment acquis le contenu contribuent une partie de celui-ci pour fournir un service de cache distribué à bénéfice de leur groupe social.

Notre contribution à ce domaine est un procédé de sélection et de codage réseau des trames en cache en fonction des préférences des utilisateurs pour les différents points de vue et les propriétés débit-distorsion du flux. Utiliser le codage réseau permet aux utilisateurs de récupérer le contenu sans échanger de grandes tables qui contiennent les emplacements des différents trames.

L'idée principale est d'exploiter les préférences des utilisateurs pour conserver dans la cache distribuée seulement les parties du contenu les plus susceptibles d'être demandés.

Nous avons comparé les performances de notre technique avec une technique de codage réseau qui ne prend pas en compte les préférences des utilisateurs, qui garde une nombre égal de trames par vue.

Nous avons observé que la mise en place des préférences, conjointement avec la contrainte imposée à la décodabilité de la sélection, améliore significativement les pérformances par rapport à la technique de référence, en termes de qualité vidéo (PSNR) pour un pourcentage donné du contenu disponible dans la cache. Cette technique et les résultats relatifs font l'objet d'un article de journal en cours de préparation.

Travaux futurs possibles comprendent le développement d'un système de distribution multi- vue interactif à grande échelle.

Plutôt que mettre en cache le contenu accessible par les noeuds, le système pourrait placer préventivement une partie du contenu dans des noeuds stratégiques, sur la base de

prévisions des demandes futures des utilisateurs. Les prévisions pourraient être déduites des préférences de noeuds proche dans leur réseau social.

En outre, bien que nous avons illustré notre approche dans le cadre MVC, il peut également être appliquée à la vidéo multi-vue avec carte de profondeur [SBB+12].

Chapitre 5 — Codage réseau a bas debit avec séparation aveugle de source

Dans le cinquième chapitre, nous discutons la possibilité de réduir l'overhead introduit par la nécessité de transmettre les coefficients de combinaison dans les techniques du codage réseau. Cela est possible si le destinataire est en mesure de reconstruire les paquets originaux à partir des mixtures sans connaître les coefficients de combinaison, une technique connue sous le nom de séparation aveugle de source (*Blind Source Separation*, BSS).

La séparation aveugle de source est le problème de la récupération d'un ensemble de signaux de source à partir d'un ensemble de mélanges observés, lorsque peu ou pas de connaissance du processus de mélange est disponible.

La séparation de source dans un corps de Galois peut trouver une application intéressante dans le contexte de codage réseau, en soulageant les noeuds de la nécessité d'envoyer les coefficients de combinaison, réduisant ainsi largement l'overhead.

Cependant, les méthodes de l'état-de-l'art, qui sont basées sur l'entropie seule, fournissent un degré insuffisant de précision pour remplacer le codage réseau pratique.

Dans ce chapitre, après un examen des travaux connexes, nous présentons deux nouvelles techniques que nous avons récemment proposé pour augmenter le pouvoir discriminant des méthodes classiques de séparation de source basée sur l'entropie.

Nous présentons nos contributions pour augmenter la capacité de discrimination des méthodes de séparation aveugle de source dans le cas de mélanges linéaires dans un corps de Galois.

Dans notre première contribution, nous avons proposé d'utiliser un codage de canal non-linéaire des signaux de source, et en particulier, on utilise un code de bit de parité impaire, qui présente l'avantage d'être très simple à mettre en oeuvre. Cependant, ces résultats peuvent être étendus au cas général d'un code de détection d'erreur non-linéaire.

Le pouvoir discriminant est augmentée dans le sens où la méthode basée sur l'entropie

sera assisté par le codage de détection d'erreur en retenant l'estimation de l'entropie seulement pour les solutions qui sont admissibles dans le sens où la source reconstruite est un mot-code. Cela élimine plusieurs solutions qui, même s'ils présentent une basse entropie et pourraient être identifiés à tort comme sources par le technique de référence, ne peut être admis comme ils ne font pas partie du code.

Nos résultats expérimentaux montrent que la technique proposée surpasse systématiquement la méthode de référence, en particulier dans le cas de sources avec une petite nombre d'échantillons disponibles, ce qui est plus critique pour les méthode basée sur l'entropie, ce qui rend la séparation aveugle de source plus approprié pour les applications pratiques, où le nombre d'échantillons est typiquement limitée par la taille d'un paquet. Cette technique a été présentée à la Conférence Internationale ICASSP 2013.

Dans notre deuxième contribution, nous avons proposé de générer, pour chaque source, résumé de message (*digest*) non-linéaire et flexible à envoyer avec les sources. Le résumé de message est généré par une fonction de hashing définie par une *sponge-construction*, ce qui permet de découpler la longueur de l'entrée et de la sortie.

En d'autres termes, la fonction est capable de générer un résumé de longueur arbitraire pour sources de n'importe quelle longueur. Le résumé de message est défini de maniere telle à être robuste vis-à-vis des combinaisons linéaires, c'est-à-dire que une combinaison linéaire de digests a très peu de chances d'être égale à la digest de la combinaison linéaire des messages correspondantes.

Cette propriété est exploitée à la destination où les observations avec un résumé non valide peuvent être jetés sans autre traitement. Sur les autres observations, qui sont un sous-ensemble beaucoup plus petit de l'espace de recherche, les méthodes traditionnelles basées sur l'entropie peuvent être appliquées.

Nos résultats montrent que cette approche améliore considérablement la capacité de séparation de techniques basées sur l'entropie dans le corps de Galois, dans les cas où les approches traditionnelles sont impossibles, c'est-à-dire, pour des sources courtes avec des distributions proches à l'uniforme.

En outre, notre technique est beaucoup plus robuste pour le problème de l'ambiguïté d'échelle, ce qui est beaucoup plus problématique dans les applications numériques que

dans la séparation aveugle de source traditionnelle analogique.

La possibilité de séparer efficacement les sources mélangées, etant donné un overhead petit et contrôlable, ouvre la possibilité pour un système de transmission similaire au codage réseau, où les sources sont linéairement combinés afin d'augmenter le débit et l'immunité face aux pertes, mais avec un overhead sensiblement réduit.

Abstract

Network coding is an innovative paradigm that allows an efficient use of the capacity of communication networks. Using network coding it is possible to maximize the throughput in a multi-hop multicast communication and to sensibly reduce the delay. Network coding has already been proposed as a solution for a wide range of scenarios: from peer-to-peer file exchange at application level, to routing at network level, and to wireless transmission at MAC/PHY level. In this thesis, we focus our attention to the integration of the network coding framework to multimedia applications, and in particular to advanced systems that provide enhanced video services to the users, such as interactive multi-view streaming. Our contributions concern several instances of advanced multimedia communications: an efficient framework for transmission of a live stream making joint use of network coding and multiple description coding; a novel transmission strategy for lossy wireless networks that guarantees a trade-off between loss resilience and short delay based on a rate-distortion optimized scheduling of the video frames, that we also extended to the case of interactive multi-view streaming; a distributed social caching system that, using network coding in conjunction with the knowledge of the users' preferences in terms of views, is able to select a replication scheme such that to provide a high video quality by accessing only other members of the social group without incurring the access cost associated with a connection to a central server and without exchanging large tables of metadata to keep track of the replicated parts; and, finally, a study on using blind source separation techniques to reduce the overhead incurred by network coding schemes based on error-detecting techniques such as parity coding and message digest generation. All our contributions are aimed at using network coding to enhance the quality of video transmission in terms of distortion and delay perceived by the users.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3DVC	3D Video Coding
ABCD	A Broadcast Content Delivery Protocol
ACK	ACKnowledgement
AVC	Advanced Video Coding
BSS	Blind Source Separation
CDF	Cumulative Distribution Function
CIF	Common Intermediate Format
DAG	Directed Acyclic Graph
EWNC	Expanding Window Network Coding
FTV	Free Viewpoint TV
GF	Galois Field
GOP	Group Of Pictures
HEVC	High Efficiency Video Coding
ICA	Independent Component Analysis
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
LCM	Linear Code Multicast
MAC	Medium Access Control
MANET	Mobile Ad-hoc NETwork
MD	Multiple Description
MDC	Multiple Description Coding

MDCT	Multiple Description Correlating Transform
MDQ	Multiple Description Quantisation
MDVC	Multiple Description Video Coding
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MRD	Maximum Rank Distance
MVC	Multi-view Video Coding
MVV	Multi-view Video
NC	Network Coding
NCV	Network Coding for Video
P2P	Peer-to-Peer network
PHY	PHYsical Layer
PNC	Practical Network Coding
PSNR	Peak Signal-to-Noise Ratio
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RLNC	Random Linear Network Coding
RTT	Round Trip Time
SD	Single Description
SDC	Single Description Coding
SNR	Signal-to-Noise Ratio
SP	Spatial Prediction
TCP	Transmission Control Protocol
UEP	Unequal Error Protection
XOR	eXclusive Or
Y-PSNR	Luminance-component PSNR

Introduction

One of the fundamental assumptions of classical networking is that a multi-hop data transfer from source to destination is handled at the intermediate nodes by forwarding the messages received on the input links to one or more of the output links, without modifying the content of the message. If more data flows share an intermediate node in their path, the intermediate node will independently assign them a priority (scheduling) and independently choose the output link through which to send them (routing).

This traditional view has recently changed with the introduction of the Network Coding (NC) paradigm. With this technique, each message sent on a node's output link is a function or mixture of the messages that arrived earlier on the node's input links. Such a strategy of mixing packets or "coding" at the intermediate nodes, together with means of decoding at the receiver, has been shown to outperform traditional routing by improving the throughput of the network and minimizing the delivery delay.

Network coding has been applied to many forms of network communications, and, in particular, the scenario in which NC has proven to offer the greatest advantage is multicasting. One important theoretical result states that, under mild assumptions on the mixing process, by only using network coding it is possible to achieve a multicast rate that fully exploits the capacity of the network.

Since this result cannot be achieved through traditional routing, NC has been successfully integrated into a wide range of applications with great benefit, such as live broadcast, distributed information storage, content delivery through peer-to-peer (P2P) networks, and interactive communications such as multimedia conferencing.

The topic of network coding has received a lot of attention from the research community, and has been approached from a multitude of disciplines, such as graph theory, information theory, channel coding theory, optimization theory, *etc.*.

In this thesis we focus on the application of the network coding paradigm to high quality video distribution services. In particular, we investigate how a network coding technique can be designed and optimized taking advantage of the knowledge of both the video coding technique used to produce the content and the rate-distortion properties of the content itself. Namely, we study the integration of network coding with advanced video coding techniques, such as multiple description coding, which is used to provide a graceful degradation in the presence of losses to the stream, and multi-view coding, which

is used to provide new and interactive 3D video services to the user.

We investigate these topics in different transmission scenarios, from wired to wireless networks, from real-time streaming to distributed caching services. For each one of these contexts, after an adequate review of the state-of-the-art, we provide our own original contributions, that have been the object of several international publications and that we validate with an analysis of our experimental results. In more detail, the rest of the manuscript is organized as follows.

Chapter 1 — Network coding

In the first chapter, we introduce the fundamental concepts of network coding and we present the most important theoretical results behind it. Furthermore, we provide an overview of the state-of-the-art, in terms of the most efficient protocols for data storage and transmission using network coding.

We provide a more detailed analysis of network coding in the context of video transmission, discussing in detail the specific requirements and constraints of video distribution. Also, we focus our attention to the application of the network coding paradigm to wireless networks, which present the additional challenge of a limited capacity and a higher loss rate.

Chapter 2 — Network coding for multiple description video

In the second chapter, we discuss the concept of multiple description coding (MDC), a coding technique meant to provide a trade-off among coding efficiency, robustness toward losses, and computational complexity. We present the basic principles of MDC, and detail how it can be used jointly with network coding to provide robustness towards errors and losses to a video stream transmitted over unreliable networks where no retransmissions are tolerated, such as the wireless ad-hoc networks.

Our first contribution is a framework to create and maintain an overlay network for a video streaming application over wireless ad-hoc networks that allows instant decoding of the received packet. The framework chooses the optimal network coding coefficients through a distributed optimization of the expected video quality using up-to-date information about the network topology provided by a protocol initially designed for real-time streaming of MDC video.

The second contribution consists in a per-hop transmission policy, based on the joint use of network coding and multiple description coding, that provides a good trade-off between loss resiliency and decoding delay. Both techniques have given promising results in terms of performance of the streaming system.

Chapter 3 — Network coding for multi-view video streaming

In the third chapter, we discuss the concept of multi-view video, an emerging video paradigm that enables new interactive services such as free view-point television and immersive teleconferencing.

In particular, we study how a network coding technique can be designed for the specific purpose of transmitting a multi-view stream. In this context, we propose our contribution: a technique based on expanding window network coding that allows immediate decoding of the multi-view stream through a rate-distortion optimized scheduling of the frame inclusions in the coding window performed at each sending opportunity.

Chapter 4 — Distributed social caching using network coding

In the fourth chapter, still in the context of multi-view video, rather than focusing on real-time transmission as in the previous chapter, we focus on a distributed caching service that, due to the high bandwidth cost of transmitting a multi-view stream, can provide a great benefit, alleviating both the server's and the network's load.

In particular, our focus is set on cooperative caching of multi-view video content, a scenario in which users who recently acquired the content contribute parts of it by providing a distributed cache service for the benefit of their social group.

Our contribution to this field is a method for selection and network encoding of the cached frames based on the users' preferences for the different views and the rate-distortion properties of the stream. Using network coding enables the users to retrieve the content without exchanging large tables maintaining the locations of different frames.

Chapter 5 — Towards low-overhead network coding with blind source separation

In the fifth chapter, we discuss the potential reduction of the overhead due to the need for transmitting the combination coefficients in network coding techniques. This can be achieved if the receiver is able to reconstruct the original packets from the mixed ones without knowing the combination coefficients, a technique known as *blind source separation*.

For our first contribution in this context, we propose to use a non-linear encoding of the source signals in order to increase the success rate of the state-of-the-art methods in blind source separation. In our second contribution we propose to append to each source a non-linear message digest, which offers an overhead smaller than a per-symbol encoding and that can be more easily tuned. Our experimental results support the possibility of using such a technique to design a transmission scheme similar to network coding that does not require the same overhead for the transmission of coefficients.

Chapter 1

Network Coding

In this chapter we introduce the theoretical aspects of *Network Coding* (NC), and its fundamental theorems and algorithms. NC can improve the performance of the network by allowing nodes to retransmit combinations of the received packets rather than mere copies.

This chapter is organized as follows. First, in Section 1.1 we introduce the basic notions concerning network coding, we review some of the fundamental theoretical results, and present the most widely used techniques for practical implementation of NC. In Section 1.2, we discuss some interesting results achieved by using network coding in the context of multimedia applications, along with some of the protocols proposed so far. Frameworks for employing network coding in a wireless scenario are presented in Section 1.3. Some improvements achievable by using a cross-layer approach are presented in Section 1.4, while some aspects of achieving error resiliency for network coding are presented in Section 1.5. Finally, Section 1.6 concludes the chapter.

1.1 Network Coding Fundamentals

In classical networking, multi-hop data transfer is handled at the intermediate nodes by relaying the received messages. The nodes forward the messages received on their input links to one or more of their output links, without ever modifying the content. If the messages belong to different data flows in the network, the node will assign them a priority (*scheduling*) and choose the output link through which to send them (*routing*).

This traditional view of network routing has been challenged by the arrival of *Network Coding* (NC) [ACLY00]. With this innovative technique, each message sent on a node's output link is a function, or mixture, of the messages that arrive on the node's input links. Such a strategy of mixing packets or "coding" at the intermediate nodes, together with means of decoding at the receiver, has been shown to outperform traditional routing by improving the network throughput and minimizing the delay.

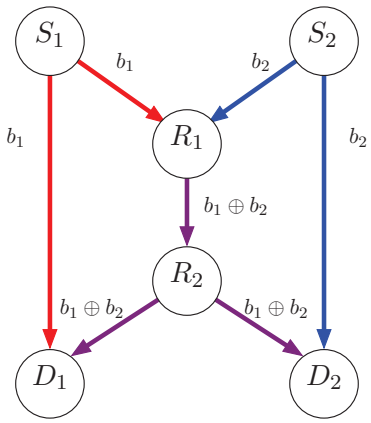


Figure 1.1: The Butterfly Network. Arcs represent directed links with capacity of 1 message per second. Source nodes S_1 and S_2 want to transmit messages b_1 and b_2 to both sink nodes, D_1 and D_2. By sending the XOR of the two messages, $b_1 \oplus b_2$, this can be achieved with a single transmission per link.

1.1.1 The Butterfly Network

The concept of NC first appeared in the seminal paper of Ahlswede *et al.* [ACLY00], introduced through the now famous example of the *Butterfly Network*, presented in Figure 1.1.

The problem of multicasting in a wireline network is considered, with two sources S_1 and S_2 wanting to deliver their respective messages b_1 and b_2 to two destination (*sink*) nodes D_1 and D_2. All links are assumed to have a capacity of one message per second. If intermediate nodes R_1 and R_2 only forward the received messages, at every sending opportunity they can either deliver both messages to D_1 but only b_1 to D_2, or both messages to D_2 but only b_2 to D_1. In other workds, the link between R_1 and R_2 becomes a bottleneck.

However, if node R_1 can send a combination of b_1 and b_2 –*e.g.*, the bitwise exclusive-or, or *XOR*– as shown in the Figure 1.1, both receivers can obtain both messages. The sink D_1 can decode b_2 by XOR-ing $b_1 \oplus b_2$ with message b_1 previously received on the direct link from S_1, and D_2 can recover b_1 in a similar way.

Network coding can thus obtain a multicast throughput of two messages/second, which is better than the routing approach that can achieve at best 1.5 messages per second.

More generally, it has been shown [ACLY00] that in the case of a point-to-point communication network in which a source multicasts information to a set of destinations, if network coding is used, a multicast capacity equal to the *min-cut* capacity of the network can be achieved, which would not be possible if only traditional routing schemes were used.

1.1.2 Max-Flow Min-Cut Theorem for Network Coding

Probably, the most important theoretical result in network coding theory is the one known as the *Max-Flow Min-Cut Theorem for Network Information Flow*, which expands the

result of the *Max-Flow Min-Cut Theorem* in graph theory [BM08].

The network can be modeled as a *direct acyclic graph* (DAG), denoted $G = (V, E, c)$, where V is the vertex set corresponding to the nodes in the network, E is the edge set corresponding to the links in the network and c are the capacities associated with the links. A simplified model considers that all links have unit capacity, which is not far from reality, as a link with integer capacity c can be replaced by c links with unit capacities.

An s—t cut in a graph is a partition of the vertex set V into sets S and T such that $s \in S$, $t \in T$, $S \cup T = V$ and $S \cap T = \emptyset$. The capacity of the cut refers to the sum of the capacities of the edges going from S to T.

An s—t cut with minimum capacity is called a minimum s—t cut and represents the minimum number of links that have to fail in order to interrupt the communication from s to t. The minimum s—t cut is an important characterization of a network, as it represents the bottleneck in the communication between the two nodes s and t.

The max-flow min-cut theorem states that in a flow network, the maximum amount of flow passing from the source s to the sink t is equal to the capacity of the minimum s—t cut. If this capacity is h, then the maximum flow in the network can be obtained using the Ford-Fulkerson algorithm [FF56], which finds the h edge disjoint paths from s to t which will carry the flow.

Ahlswede *et al.* point out that in order for a source to multicast information at a rate approaching the smallest minimum cut between the source and any receiver, some sort of network coding has to be allowed at intermediate nodes. Codes usually used in source/channel coding can be easily employed for network codes.

A data unit can be represented by an element of a base field F_q, with q the size of the field. In the butterfly network, F is the finite field (or *Galois Field*) GF(2). A message consisting of h data units can be represented as a vector $x \in F_q^h$. The propagation of message x in the network is represented by the transmission of a symbol $f_e(x) \in F_q^h$ over any channel/link e in the network. The encoding mechanism for every channel specifies the network code. With network coding, the max-flow in the network can be obtained as long as the symbols are coded in a finite field F_q whose size q is large enough.

1.1.3 Linear Network Coding

This max-flow min-cut theorem for network coding was first presented as a conjecture, but subsequent work [LYC03]) showed that, in a directed acyclic graph, the multicast capacity can be achieved, and it suffices for the encoding functions to be linear. Also, a finite field size can be chosen for the symbols. The proof is in fact a constructive algorithm for the encoding functions, named a generic *Linear Code Multicast* (LCM). If the links are ordered in topological order, then the encoding vector assigned to the current link should be linearly independent with respect to all previously assigned coding vectors.

The next step in investigating the innovative technique of network coding was to for-

mulate and solve the multicast problem, *i.e.*, finding necessary and sufficient conditions so that a given set of connections can be achievable over a given network. This was achieved by Koetter and Médard by introducing an algebraic framework for linear networks [KM03]. Their result is a formulation of the feasibility of a multicast problem and the validity of a network coding solution in terms of transfer matrices. The result was also extended to arbitrary networks.

The algebraic framework was based on the observation that, if the network is linear over F_2^m, the relationship between an input vector $\mathbf{x}$ and an output vector $\mathbf{z}$ can be described by a transfer matrix $\mathbf{M}$: $\mathbf{z} = \mathbf{xM}$, where $\mathbf{M}$ has coefficients from F_2^m. An outline of the results is given in the following.

When the network is represented as a directed graph G having the set of vertices V and the set of edges E, a topological ordering is an ordering of nodes $v_1, \ldots, v_n$ such that if edge $(v_i, v_j) \in G$ then $i < j$, *i.e.*, an edge can only point from a node with lower index to one with higher index. Any graph can be topologically ordered by renaming the vertices. Then the adjacency matrix of the edges in the network, the $|E| \times |E|$ matrix $\mathbf{F}$, will be upper triangular.

If there are μ sources in the network and ν sink nodes, then the transfer matrix $\mathbf{M}$ can be expressed in terms of matrices: $\mathbf{A}$ (of dimension $\mu \times |E|$, that specifies the transformation from the sources to the edges of the network), $\mathbf{F}$ (the adjacency matrix, of dimension $|E| \times |E|$) and matrix $\mathbf{B}$ (of dimension $\nu \times |E|$ that specifies the transformation from the edges of the network to the outputs) in the form:

$$\mathbf{M} = \mathbf{A}(\mathbf{I} - \mathbf{F})^{-1}\mathbf{B}^\top. \tag{1.1}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ specify how the data enter and leave the network. Matrix $(\mathbf{I} - \mathbf{F})^{-1}$ should specify how the data are propagated through the network, *i.e.*, how the symbol sent on each edge contributes to the symbols sent on other edges in the network. This is true since, for any pair of edges (e_i, e_j), the contribution of e_i to the linear combination carried by e_j is given by all possible paths that start from e_i and end at e_j. The paths are captured in the sum $\mathbf{I} + \mathbf{F} + \mathbf{F}^2 + \cdots + \mathbf{F}^N$, where N is the longest path in the network.

As previously mentioned, if the graph is topologically ordered, $\mathbf{F}$ is upper triangular and also $\mathbf{F}^N = 0$. Thus, since $(\mathbf{I} - \mathbf{F})\big(\mathbf{I} + \mathbf{F} + \mathbf{F}^2 + \cdots + \mathbf{F}^N\big) = \mathbf{I} - \mathbf{F}^{N+1} = \mathbf{I}$, it follows that $(\mathbf{I} - \mathbf{F})^{-1} = \big(\mathbf{I} + \mathbf{F} + \mathbf{F}^2 + \cdots + \mathbf{F}^N\big)$.

If matrices $\mathbf{A}$ and $\mathbf{B}$ have the same dimensions, with h the minimum cut in the network, the output vector $\mathbf{z}$ can be recovered if the transfer matrix is invertible, *i.e.*, $\det(\mathbf{M}) \geqslant 0$. It can also be shown that:

$$\det(\mathbf{M}) = \pm \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{I} - \mathbf{F} & \mathbf{B}^\top \end{bmatrix}.$$

The *Multicast Theorem of Network Coding* is restated as: given a directed acyclic graph G with unit capacities, that has a single source node s (with h messages) and

a set of terminal nodes T, the multicast property with rate h is said to be satisfied if $max - flow(s, T_i) \geq h$; for all $T_i \in T$. If G satisfies the multicast property, a network code that supports the multicast rate h is guaranteed to exist as long as the field size is larger than $|T|$ [KM03].

Proof. For all the terminal nodes T_i to be able to recover the h source messages, the following has to hold: $\prod_{i=1}^{|T|} \det(\mathbf{M}_i) \neq 0$.

If the coefficients of matrices $\mathbf{A}$, $\mathbf{B}^\top$, and $\mathbf{I} - \mathbf{F}$ are denoted by α, β and ε, then the product of determinants is a polynomial in variables α's, β's, and ε's.

Since each variable in the right-side matrix of Equation 1.1 appears at most once, then in the determinant each variable also appears at most once. Therefore, in the product of determinants the maximum degree of any variable is at most $|T|$. Using the sparse zero's lemma (Schwartz-Zippel Lemma [Sch80]), one can show that an assignment of the variables such that the polynomial evaluates to a non-zero value exists as long as the size of the field is greater than $|T|$. □

Subsequent work dealt with finding algorithms capable of solving the multicast problem and that could be implemented in practice. Sanders, Jaggi *et al.* [JSC+05] considered networks in terms of acyclic delay-free graphs and studied the single-source multicast problem. They provided centralized deterministic and randomized polynomial-time algorithms for finding network coding solutions by considering subgraphs consisting of flows to each receiver. If the maximum rate allowed in the single-source multicast scenario is h, the algorithm first finds for each receiver t the s—t flows in the network. As mentioned above, this can be done with a classical Ford-Fulkerson algorithm [FF56] that finds the h edge-disjoint paths from the source to each receiver. Due to the fact that many s—t flows will overlap, the number of required operations is reduced compared to LCM by assigning coding vectors only to the links serving multiple flows. A detailed presentation of the *Linear Information Flow* (LIF) algorithm can be found in [JSC+05].

1.1.4 Practical Network Coding

Coding strategies like the one previously presented on the butterfly network imply that there is certain knowledge of the network topology and that dedicated nodes are responsible for performing the same encoding operations. The scheme is thus centralized and fixed.

In real networks however, the structure, topology and traffic demands may change quickly and drastically, while the information about those changes propagates with a certain delay. In wired networks, the edge capacities may vary due to changing traffic conditions and congestion. In peer-to-peer overlay networks many nodes may join or leave the network in a short interval. A wireless network may vary in time due to fading channels,

interference and node mobility. In wireless ad-hoc networks, where the nodes are self-organizing, the participating nodes also have limited resources in terms of communication and computation, and transmission quality may vary due to node mobility.

In the approaches presented so far the network conditions are considered fixed over a fairly large period of time, but in practice each change in the network would imply computing new optimal combination operations. In networks with cycles and delays, network codes can be obtained in a distributed manner, by performing *Random Linear Network Coding* (RLNC) [HMS+03], *i.e.*, the coefficients of the linear encoding functions are chosen independently and randomly from the finite field.

In the standard framework, as previously explained, the network is modeled by an acyclic graph $G(V, E)$ having unit capacity edges E, a source $s \in V$ and a set of receivers T. The multicast capacity h is the capacity of the minimum cut between the source and any of the receivers. The h source messages are denoted by $x_1, \ldots, x_h$ and are symbols over a field F_q where the order (size) of the field q is finite.

The probability that a set of coefficients chosen randomly from the finite field F_q does not ensure decodability at the receiver is given by: $p_e = 1 - \left(1 - \frac{|T|}{q}\right)^{|E|}$. This result shows that the error probability in the case of random network codes does not depend on the maximum multicast rate h. Moreover, by working with a field size q large enough, the probability of error can be made negligible. The problem then rests in transmitting the coding operations that take place with each packet.

A practical approach was presented by Chou *et al.* [CWJ03]. Since then, it has become –under the denomination of Practical Network Coding (PNC)– the de-facto method for random linear network coding, and has been successfully applied to solve a series of problems in multimedia applications, described in Section 1.2. We briefly present it in the following.

Any intermediate node $v \in V$ will have a set of incoming edges $\Gamma_{in} = \{e' | out(e') = v\}$ and a set of outgoing edges $\Gamma_{out} = \{e | in(e) = v\}$, as depicted in Figure 2. Each edge $e \in E$ going out of node v carries a symbol $y(e)$, which is computed as a linear combination of the symbols $y(e'_i)$ on the incoming edges e' of node v, *i.e.*, $y(e) = \sum_{e' \in \Gamma_{in}} m_e(e') y_e(e')$ with $m_e(e') \in F_q$. The coefficient vector $\mathbf{m(e)} = [m_e(e')]$ represents the local encoding vector along edge e.

By induction, the symbol $y(e)$ on any edge can be computed as a linear combination of the source symbols, *i.e.*, $y(e) = \sum_{i=1}^{h} g_i(e) x_i$.The coefficients form a global encoding vector $\mathbf{g}(e) = [g_1(e), \ldots, g_h(e)]$, which is updated at each coding operation using the local encoding vectors $m(e)$. The global encoding vector $\mathbf{g}(e)$ represents the code symbol $y(e)$ in terms of the source symbols $x_1, \ldots, x_h$.

When a sink node $t \in T$ receives symbols $y(e_1), \ldots, y(e_h)$, they can be expressed in

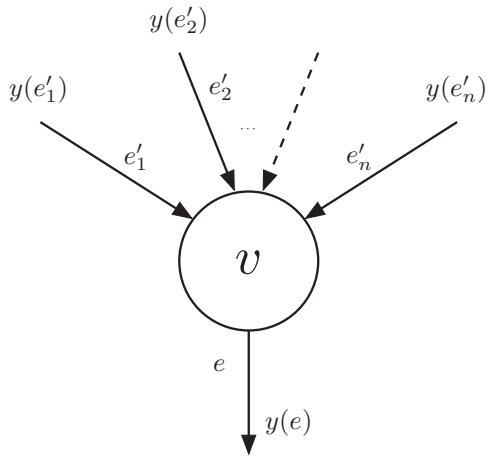


Figure 1.2: Intermediate node in a network with coding capability. $y(e'_i)$ represent the symbols carried by incoming edges e'_i in node v, while $y(e)$ is the symbol transmitted on outgoing edge e by the node after the encoding function.

terms of the source symbols as:

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

where G_t is the global encoding matrix and the i-th row of G_t is the global encoding vector corresponding to $y(e_i)$. Sink t can recover the h source symbols as long as G_t is invertible, *i.e.*, has full rank h.

This is true with high probability as long as the coefficients of matrix G_t are chosen uniformly at random from a field of sufficiently large size q [HMS+03]. Simulation results [CWJ03] show that even small field sizes (Galois Field GF $\left(2^8\right)$ or GF $\left(2^{16}\right)$) work well in practice and the probability of error becomes negligible.

To generalize, each packet in the network can be considered as a vector of symbols. The above algebraic relationships remain valid for packets if the source symbols are grouped into packets. In order to be able to invert the code at any receiver, the global encoding vectors should be obtainable from the arriving packets themselves. This is done by prepending the i-th source packet $\mathbf{x}_i$ with the i-th unit vector.

$$\begin{bmatrix} \mathbf{y}(e_1) \\ \vdots \\ \mathbf{y}(e_h) \end{bmatrix} = G_t \begin{bmatrix} 1 & \cdots & 0 & \mathbf{x}_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & \mathbf{x}_h \end{bmatrix}$$

The benefit of these tags is that the global encoding vectors can be found within the packets themselves. The sinks can compute G_t without knowing the network topology or packet-forwarding paths. The network can thus be dynamic, with nodes and edges being

added or removed in an ad-hoc way.

In practice the data stream is divided into generations with each generation consisting of h data packets. Intermediate nodes only mix packets coming from the same generation. The source generates h new data packets by prepending the h-dimensional i-th unit vector to the corresponding i-th data packet. Intermediate nodes store the received packets into buffers sorted by generation number and when a transmission opportunity occurs, the packet to be sent is generated by random network coding of all the packets in the current generation. The original packets are obtained at the receivers by performing Gaussian elimination as soon as h independent coding vectors have been received.

Since, in order to decode, a sink node has to wait for a number of independent packets equal to the generation size, this parameter clearly affects the delay in the network, so it is desirable to keep it as small as possible. Also, the generation size gives the size of the overhead of transmitting the coefficients in the packet header, and it should also be kept small. As an example, an IP packet over the internet has a maximum size of 1500 bytes. If every byte is treated as a symbol over GF $\left(2^8\right)$ and the generation size is $h = 50$, then the overhead is approximately $\frac{50}{1500} = 3.3\,\%$. This aspect might become important for wireless communications where the packets are smaller and the overhead may become important. On the other hand, the size of the generation affects how well packets are "mixed". The decoding matrix would be full rank with a probability at least $\left(1 - \frac{1}{q}\right)^h$, where h is the generation size, so it is desirable to have a fairly large generation size.

1.2 Network Coding for Multimedia Applications

In the previous section, the subject of network coding has been approached from a graph and information theory perspective. From the networking perspective, the OSI or the TCP/IP stack protocols offer several options regarding the level where NC should be implemented. In content distribution networks NC can be implemented at the application layer with little effort. In self-organizing networks it can be used for the wide dissemination of important data, as a substitute for routing at the network layer. In general packet networks there is a high interest in integrating NC with the current TCP/IP protocol, while investigating the coding opportunities at the physical layer in wireless networks can be an interesting topic of research. For multimedia applications though, the recent trend is to build protocols that are implemented as a cross-layer between the application and network layer, so that they can be tailored to the specific media content to be delivered.

The NC approaches can also be divided in two broad categories. If coding is allowed only among packets belonging to the same multicast session, those problems are also known as intra-session network coding and are based on PNC. The other category, which includes the butterfly network, allows for the combination of packets from different multicast or unicast sessions, thus it is referred to as inter-session network coding. In this section we

discuss some of the existing NC approaches for multimedia applications.

1.2.1 NC-based Distributed Storage Systems

Distributed storage systems are a solution for the next generation multimedia applications that require increased storage space, ease of access and reliable recovery. As an example, more and more companies offer cloud services for distributed computing and storage. In order to allow information to be spread over multiple, unreliable nodes situated at different locations, such systems have to ensure the redundancy needed for reliable recovery in case of node failures. Hence, the bandwidth a replacing node requires in order to obtain the information needed for the reconstruction of lost data, also known as the repair bandwidth, is an important parameter of the system. Other parameters that influence the system's capability to recover, such as the delay in access and transmission, can be reduced by taking into account the geographical distribution of resources.

In a distributed storage system the storage elements are placed into nodes which function independently of each other and thus exhibit independent failure patterns. These nodes are often connected through a network with arbitrary topology, and the information objects are stored in specific nodes according to a mapping function.

The use of NC for distributed storage systems has been studied by Dimakis, Wu *et al.* in several papers [DGW$^+$10, Wu10] and has been compared to the results obtained by using traditional erasure codes. Since in storage systems that are distributed over networks the nodes may fail or leave the system quite often and the codes have to be maintained over time, the problem rests in finding a trade-off between the storage capability and repair bandwidth.

If in the system an (N, M) erasure code is used, each information object is divided into M distinct blocks which are encoded using appropriate erasure codes like the popular Reed-Solomon codes. Encoding of the M original blocks generates $N \geq M$ encoded blocks and the entire object can be recovered from any $M\gamma$ out of the total N pieces, with $\gamma \geq 1$. The code has a rate $R = M/N \leq 1$, and the relative redundancy introduced equals $(N - M)/M$.

If the encoded blocks are stored at different nodes in the system, when a node fails, recovery can be obtained if a new node downloads subsets of data stored at a number of surviving nodes and uses them to reconstruct the lost data. The data would not be able to be recovered only if the number of nodes that fail is greater than the introduced redundancy. The amount of storage needed to obtain a similar resilience by means of simple replication would be prohibitive. However, erasure coding and in general coding-based methods achieve best results in scenarios with medium/low node availability, as in mobile mesh or ad-hoc networks, and the complexity introduced by maintaining these codes is quite high.

Although erasure codes offer a good trade-off between redundancy and error tolerance,

in practical storage systems a critical resource is also the network bandwidth. Minimizing the network bandwidth needed by repair operations is another important aspect to be considered when designing the code.

In order to formulate the repair problem, Dimakis *et al.* [DGW+10] introduce the information flow graph to represent the evolution of information flow as nodes join and leave the system. Each storage node is represented in the flow graph by a pair of nodes connected by an edge whose capacity is the storage capacity of the node. A source s is supposed to have the entire file. The requests to reconstruct the original data from a subset of nodes are called data collectors and are represented as sink nodes whose input edges have infinite capacity.

Through the information flow graph, the original storage problem can be formulated as a network communication problem, where the source s multicasts the file to the set of all possible data collectors. Performance bounds for codes can be derived by analyzing the connectivity in the information flow graph. Network coding can be employed over the resulting multicast graph to achieve the max-flow of the network, as described in the previous section.

If the repair operations are posed as a max-flow min-cut problem between source and any arriving node in the induced connectivity graph, repair bandwidth requirements can be derived so that the min-cut k of the graph would suffice to eventually recover the stored objects by just querying k storage nodes. However, if the minimum cut k between s and a data collector t is less than the size of the original file, then it is impossible for the data collector to reconstruct the original file by accessing only k storage nodes, regardless of the code used.

1.2.2 P2P-based Content Distribution

Intra-session network coding based on PNC has been proven beneficial for large scale content distribution in peer-to-peer (P2P) overlay networks [GR05]. In such a network the server splits the file into small blocks or chunks which are downloaded by end-users in parallel from different nodes. Once a user has downloaded a given block, that device can act as a server (*seed*) for anyone else interested in that file.

The most popular of such cooperative architecture, BitTorrent, uses a rarest-first block download policy. It attempts a uniform distribution of parts among the nodes to prevent users who have all but a few pieces from waiting too long to finish their download. However, some blocks often remain "rare", so when nodes are close to finishing their download, they may attempt to obtain them from the server, causing unnecessary server overloading. Other inefficiencies in traditional P2P systems are more pronounced in large heterogeneous networks during flash crowds, in environments with high churn –*i.e.*, many peers connecting or disconnecting in a short period of time– or when cooperative incentive mechanisms are in place.

The problem of block scarcity can be solved by allowing nodes to send a linear combination of all their available blocks. A first advantage of this network coding scheme is that the system becomes more robust to situations when the server and/or the nodes leave the system. Due to the uniform random distribution of coefficients in the linear combinations, all nodes are able to finish the download, even in extreme situations.

A second advantage can be observed when scheduling the blocks for transmission. In a large scale network that does not have a central scheduler, optimal scheduling of packets is difficult to achieve, and nodes usually have to make a decision based on local information only, which can be suboptimal. Using network coding, this decision is greatly simplified. A node can determine if it can provide an innovative packet to a neighbor by comparing their coefficient matrices. However, if this information is not available to the sender, it can generate a linear combination of all the coefficient vectors previously received and send the resulting vector. The receiver can afterwards check if the received vector is linearly independent with its own coefficient vectors and thus it can determine if the sender can provide new blocks.

The first protocol that uses PNC in P2P networks was Avalanche [GR05]. In order to function as a complete P2P system, Avalanche has three types of participants: peers, registrars and loggers. A peer can be a source or sink for the content to be distributed; if it sends content into the system, then it is called server. If a peer remains in the system after it finished downloading, then it becomes a seed. Registrars and loggers are responsible for the overlay management. A registrar enables peer discovery and provides nodes with a set of active peers. Every peer reports to the registrar when it needs more neighbors and also reports detailed statistics to the logger. Each peer maintains 4–8 connections to other peers, out of which a neighbor is periodically dropped at random to prevent formation of isolated clusters.

Avalanche has been compared with schemes where information is sent unencoded and to schemes where only the server generates and transmits encoded packets. In clustered topologies, network coding shows a clear benefit. In such scenarios, since peers may belong to different clusters, without coding, some packets travel several times over the links that connect the clusters, thus wasting capacity. In heterogeneous networks, where nodes have different upload and download capacities, the performance achieved by the fast peers is degraded without coding. This is caused by slow nodes that may spend their bandwidth downloading from the server blocks that are not useful to fast peers.

Live P2P streaming is considered in [WL07a], where results are based on an experimental testbed consisting of a cluster of 44 dedicated dual-CPU servers. In their framework, Lava, a node has two functionalities: a network part that deals with the connections and emulates the upload and download capacities, and the algorithm part that implements the algorithms and protocols for live P2P streaming. A live session consists of a multimedia stream with a specific multimedia rate, each stream being divided into segments further divided into blocks. In order to evaluate the benefits of using network coding when

compared to traditional protocols, the network coding algorithm was added as a plugin component to a standard P2P protocol called Vanilla. Nodes keep a playback buffer in which segments for a streaming session are stored in order and removed after they have been played. If the segment is not available in time for playing it will be skipped. A player starts to produce and serve new coded blocks after it receives n coded blocks, where n is the number of blocks in a segment and $0 \leq a \leq 1$ is a parameter called aggressiveness which can be tuned in order to evaluate the performance of the system. As a measure of system performance the authors use the percentage of playback skips and of discarded blocks, as they are given by the number of linearly dependent packets and by the number of obsolete packets in the system.

Their result show that network coding can support a wide range of streaming rates, but the decoding process represents a bottleneck in the streaming process. In the flash crowd scenario network coding shows a higher percentage of initial playback skips but is more resilient to peer departures. Subsequent work by the authors [WL07b] shows that better throughput can be achieved with a push-based protocol that uses information exchange among peers before generating a new coded packet. Other architectures that use peer information exchange are discussed in Section 1.3.

1.3 NC for Multimedia Applications in Wireless Networks

Wireless technologies have been adopted quickly thanks to their advantages in terms of mobility, low cost equipment and ease of deployment compared to wired networks. In addition to maximizing throughput, network coding can offer other advantages in wireless networks, as it can be used to reduce the energy consumption by reducing the number of transmissions.

The advantages of NC are more evident in wireless ad-hoc networks, as in the simple scenario of two nodes wanting to exchange messages through one relay node presented in Figure 1.3. Each of the nodes A and B safely transmits its message to the relay R which will afterwards broadcast the XOR of the two messages. This scheme allows each node to decode the desired message after three time slots as compared to the traditional four time slots. This is therefore an important result, as the energy efficiency –the amount of battery energy consumed to transmit bits across a wireless link– is a critical design parameter for wireless ad-hoc networks.

A wireless ad-hoc network is a decentralized type of wireless network that, unlike traditional wireless networks, does not rely on access points for management. Each node participates in routing by forwarding data for other nodes, and the decision on which nodes to forward is made dynamically based on the network connectivity. In wireless mesh networks, a packet travels multiple wireless links before reaching a gateway node, which is connected to the wired Internet. Mesh networks can be preferable to single-hop access point networks as they can achieve the same coverage either with much lower transmission

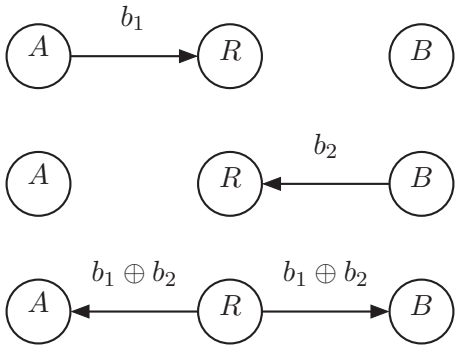


Figure 1.3: Network coding for message exchange in one-hop wireless network. The number of total transmissions in reduced from 4 to 3 if the relay node R broadcasts a XOR of the two messages received from nodes A and B.

power or with lower deployment costs. The wireless routers in a mesh network form the backbone for clients. Clients can also act as routers and participate in forwarding packets, although typically they are end-user nodes with simpler wireless hardware and software. A network in which all nodes can transmit, receive as well as relay data and where the topology may change over the course of time and space is called a *Mobile Ad-hoc Network* (MANET) [FJL00].

Although wireless ad-hoc networks should be by definition cooperative, so far the communication has been implemented as point to point communication, similar to wireline or traditional access-point wireless networks. With current protocols, even though a sent packet is in fact broadcast and can reach all nodes in a neighborhood, the unintended recipients have to drop the packet. By allowing intermediate nodes to store unintended packets and then use them in network coding combinations, the throughput of the network can be increased, both in the case of multicast [WFLB05] and multiple unicasts [KRH$^+$08].

In the case of multicast, the participation in the forwarding process of a large percentage of nodes in the network and the use of random linear network coding can achieve high delivery of packets even when the packet loss rates are high [WFLB05]. This makes NC much more energy efficient in wireless ad-hoc networks and could replace flooding, currently used for rapid dissemination of important messages in the network.

The first NC protocol for wireless scenarios was the COPE protocol, proposed by Katti *et al.* [KRH$^+$08] and it uses simple binary XORs as its network coding operations in wireless mesh networks. COPE takes advantage of the broadcast nature of the wireless medium and allows nearby nodes to overhear a packet, which means a node can place the packet in its buffer even though the node is neither the destination for that packet nor a forwarding node in that particular transmission. More nodes can therefore retransmit the overheard packet or combine it with other packets if this would be beneficial for other nodes in the network. COPE is designed to perform network coding for multiple wireless unicast flows, which raises other problems than the multicast cases studied so far. In unicast, packets from multiple unicast flows may get encoded at some intermediate node,

but the paths will diverge later, at which point the flows need to be decodable. COPE also deals with many technical issues that may appear when implementing NC in today's wireless networks, shortly presented in the following.

Each node stores the overheard packets for a short time and also tells its neighbors which packets it has heard by annotating the packets it sends or by periodic exchange of status reports. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform opportunistic coding; the node XORs multiple packets and transmits them as a single packet if each intended next-hop neighbor has enough information to decode the encoded packet. This extends COPE beyond the case of two flows traversing the same node, as it allows to XOR more than two packets.

In the wireless environments, the main benefit of NC is that it the encoded packets are broadcast and meant to be received by all neighboring nodes. However, the 802.11 broadcast has no collision detection or avoidance mechanism, as it requires no acknowledgement for the safely delivery, in contrast to the unicast case. The 802.11 unicast mode requires unicast packets to be immediately acknowledged by their intended next-hop. The lack of an ACK is interpreted as a collision by the MAC layer which will react by backing off and allowing other nodes to share the medium before attempting to retransmit. In the broadcast case, the packet has many intended receivers and an ACK from all of them would add too much overhead to the communication. Rather than changing the MAC layer completely, the COPE protocol tries to work with the existing 802.11 standard, by using pseudo-broadcast.

In pseudo-broadcast, the packets meant to be broadcast are actually sent as unicast packets with the link layer destination field set to the MAC address of one of the intended receivers. An extra header is added after the link layer header in which all the intended recipients are listed. Since nodes work in the promiscuous mode, which allows them to check packets not addressed to them, they can find out if they are among the intended recipients, while only the target node sends the ACK. By using the 802.11 unicast, the MAC layer can detect collisions and can backoff accordingly.

The results of using a NC protocol in wireless environments clearly show an improvement in network throughput, particularly as the number of flows increases. However, even though with the increase of network flows the coding opportunities increase, the congestion also increases and many reception reports may get lost. The best results are obtained by COPE together with opportunistic routing. This implies that an intermediate node coding packets together checks not only the reception reports but also the path chosen for the packet by a routing protocol. The path is added to the header of the packet and if the node discovers that it is a forwarding node for the packet, it takes responsibility for retransmitting it.

While NC ensures earlier delivery of packets and reduces the number of transmissions in the network, improvements to current protocols can be made.

1.4 Cross-Layer Optimization

So far, the problem of network coding has been translated into improving the network throughput, *i.e.*, ensuring that the received packets are innovative for the destination nodes. Most works previously presented have been proposed at the network layer, concerned with finding paths and assigning network codes for those paths so that more of the transmitted packets can be decoded by most receivers. In real networks however the quality of service and the amount of traffic are an important issue, which is why a network coding algorithm should take into account the relative importance of media packets and prioritize their delivery. Other works have tried to deal with the problem at the source, by adding different protection strengths to different priority levels, but without knowledge of the network conditions. The works presented in this section have shown that a cross-layer optimization approach achieves better performance in lossy networks such as overlay networks or multi-hop wireless networks.

In the wireless case, several algorithms build on the COPE framework, described in Section 1.3. In COPE, several packets from different unicast streams are packed into a single code for transmission that can be decoded at the next-hop node, thus ensuring the increase in network throughput. In addition to that, the network code should also take into account the importance of packets –in terms of contribution to the overall quality and playout deadline– within the same stream. The combined approach from [SM09] is presented in the following, as it has been shown to improve the quality of video delivery while maintaining the same level of throughput as the COPE protocol. Although the algorithms were proposed for video streams, they can be extended to any type of media applications.

The pseudo-broadcast is implemented similar to the COPE protocol, on top of the 802.11 unicast, and nodes can learn about the contents of the virtual buffers of their neighbors either explicitly through periodic status reports or implicitly from the annotations in the packet headers. NC is implemented as a thin layer between IP and MAC, as in the original COPE.

The algorithm has to first solve the code construction problem, *i.e.*, deciding which packets can be coded together, and then the code selection problem, *i.e.*, choosing a code that improves the quality with respect to a certain metric, in this case the video quality. The intermediate node maintains a transmission queue with incoming video packets, and at a given time slot a packet is chosen for transmission, called the primary packet, whose destination is called the target node. The primary packet is XOR-ed together with other side packets that can be useful to nodes other than the target node. The coded packet is broadcast to all nodes in the neighborhood. The target node will be able to decode the packet due to the code construction and will broadcast an acknowledgement (ACK). The other neighboring nodes will overhear the packet and store it in their virtual buffer until an ACK from the target node is received or until the packet deadline expires. From

the time it sends a network code the intermediate node will wait for a mean round-trip time (RTT) to receive the ACK, during which the packets that were part of the code will be marked as inactive in the transmission queue, making them available as side packets but not as primary packets. When an ACK is received, the primary packet is removed from the transmission queue. When the RTT time expires without receiving an ACK, the packet is marked as active again and the node will try to retransmit it until the deadline expires.

The first algorithm, called NCV, constructs all candidate network codes that include the primary packet, among which it chooses the code that maximizes the total video quality improvement. In order for the network code to be optimal in a rate-distortion manner, the distortion value of every packet is determined at the source and communicated to the intermediate nodes in the packet header. In addition to the importance of packets in the flow (Δ), the different flows may be of different importance (γ), so the total importance of the packet is considered to be the product of the two terms, $\gamma\Delta$

By construction, the network code corresponding to the primary packet p_i has to be decodable by the target node, which means the side packets must be among the packets already decoded at the target node, but still useful to other nodes. For each possible code c_k, the improvement of video quality at each neighboring node n is given by:

$$I_k(n) = \sum_{l=1}^{L_k} (1 - P(l))\Delta(l)\gamma(l){g_l}^k(n){d_l}^k(n)$$

where L_k is the number of original packets included in the code (packets that can be useful to any node, while the target node can benefit from only one such packet), ${g_l}^k(n)$ and ${d_l}^k(n)$ are indicator functions that take the value 1 if code c_k is useful for node n and 0 otherwise, $\Delta(l)$ is the improvement in video quality if packet l is received correctly and on time at client n, $\gamma(l)$ is the priority of the flow the packet belongs to, $P(l)$ is the loss probability of the packet, due to either channel errors or late arrival.

For every primary packet p_i, NCV chooses the code c_k that maximizes the video quality improvement at all clients. NCV selects as primary packet the active packet at the head of the queue, which in turn determines the optimal selection of side packets. To optimize the selection of primary packet, the NCV algorithm is extended to looking into the queue in Depth (NCVD), *i.e.*, considering all packets in the queue as candidates for the primary packet. Although this algorithm increases the options for candidate codes, it also requires more computation for code construction and selection.

The authors further integrate NC with the well-known rate-distortion optimized packet scheduling framework (RaDiO). Without NC, in classical RaDiO packet scheduling, the node would choose a policy π for the next transmission opportunity. For every packet π in the queue, the policy would indicate whether this packet is transmitted $\pi(j) = 1$ or not $\pi(j) = 0$, so as to minimize a weighted function of distortion and rate $J(\pi) = D(\pi)+\lambda R(\pi)$.

With NC, the goal is to find the optimal code transmission policy on all nodes Π_{valid}, so as to minimize the total distortion D_{valid}, subject to the rate constraint $R(\Pi_{valid}) \leq R_{av}$, where R_{av} is the available bit rate. Using Lagrangian relaxation, the problem turns to finding the code transmission policy Π_{valid} such that $J(\Pi_{valid}) = D(\Pi_{valid}) + \lambda R(\Pi_{valid})$ is minimized. Instead of finding the optimal code transmission policy, each code can be mapped into the packets it contains and the problem can be turned into finding the optimal packet transmission policy.

The equivalent problem is to choose the packet policy π such that:

$$\min_{\pi,\lambda} J(\pi) = \min_{\pi,\lambda} D(\pi) + \lambda R(\pi),$$

where $D(\pi)$ and $R(\pi)$ are the total distortion and rate over all nodes under the policy π.

The optimal policy decides which node n should transmit and what code c_u should be transmitted by choosing the maximum Lagrange multiplier $\max_{n,c_u} \lambda_n(c_u)$. In practice this can be achieved in two rounds: first, each node n compares $\lambda_n(c_u)$ for all possible codes and finds $\lambda_n = \max_{c_u} \lambda_n(c_u)$; then all nodes exchange their λ_n values with their neighbors. Finally, the node with the maximum λ is the one who transmits and this is repeated at each transmission opportunity. Although this method obtains the global optimum at each transmission, it cannot be implemented in practice, as it requires either complete knowledge of the network, or the exchange of messages among all nodes. Simulations show however that the previously presented algorithms, NVC and NCVD, can perform well in practice with less message exchange and that they can be considered efficient heuristics to the general NC-RaDiO problem.

For overlay networks, a receiver-driven video streaming solution is proposed [TCF11] for video packets belonging to different priority classes. The problem of choosing the network coding strategy at every peer is formulated as an optimization problem of determining the rate allocation between the different packet classes such that the average distortion at the requesting peer is minimized.

The packet classes can correspond to layers in scalable video streams or can be constructed based on the contribution of each packet to the overall quality of the media content. Class c is defined as the set of packets that are linear random combinations of packets from the c most important classes. The class number is identified in a small header in each packet. The protocol follows two stages: first, children nodes u_i compute the optimal coding strategy that their parents v_j should follow based on the available bandwidth, importance of packets in each class, and expected loss probability of the link. They then send a request message to their parents specifying the number of packets they want to receive from each class. Parent nodes send random linear combinations of packets in the requested classes. Based on the state of its buffer and the local network status, the child node recomputes the optimal coding strategy and makes another request. In this way the algorithm is receiver-driven and can adapt to the needs of each node and to changing

network conditions.

A child node u sends the same request to all its parents, which takes the form of a rate distribution vector $w = [w_1, \ldots, w_C]$, where w_c represents the proportion of packets from class C in the requested packets, so $\sum_{c=1}^{C} w_c = 1$ and $w_c \geq 0$. The expected reduction in video distortion $D(u)$ is a function of the number of classes the node u can decode, and can be written as:

$$D(u) = \sum_{c=1}^{C} d_c p_d(c),$$

where $p_d(c)$ is the probability that node u is able to decode c video classes, *i.e.*, the probability that it receives enough packets to decode packets up to class c, but not the subsequent classes.

Each client node thus solves a Rate Allocation Problem (RAP) which is formulated as finding the optimal w^* distribution over the classes that minimizes the expected reduction in distortion:

$$w^* = arg \max_{w} D(u)$$

such that $w_c \geq 0$ and $\sum_{c=1}^{C} w_c = 1$, for $c = 1 \ldots C$.

The authors further show that the optimization function can be put in a log-concave form that can be solved by means of iterative algorithms used in convex optimization problems. This aspect is important since every client has to solve the RAP problem independently and the search space would be huge if they were to do an exhaustive search. The authors also propose a greedy algorithm that is able to find a solution in a finite number of steps.

Experimental results for such cross-layer optimization schemes show that they can not only improve the quality of the service provided, but they adapt well to different network characteristics like size of the network, link capacity or packet loss probability.

1.5 Error Resilient Network Coding

Network coding is based mostly on performing linear coding operations at intermediate nodes. If each sink node is aware of both the coding functions and the network topology, perfect decoding is possible by solving a system of linear equations provided that no errors have occurred in the network. However, the assumption of error-free networks is problematic since various kinds of errors are likely to take place in real networks. In a wireless scenario, for instance, packets may experience random errors due to noisy links. Furthermore, malicious nodes may intentionally inject corrupted packets in order to alter information packets. Since even a single error has the potential to affect the decoded messages at all sink nodes, methods presented in the previous sections perform network coding at the application or network layer, after the erroneous packets have been dropped at the MAC layer. However, the transmission efficiency could potentially be improved by

employing error-correcting codes and thus avoiding retransmission.

The problem of error-control in random linear network coding was considered by Koetter and Kschischang [KK08] starting from the observation that linear network coding is vector space preserving. The original data (packets) injected by the source is modeled as a basis for a vector space V and the network itself is considered as a black box, *i.e.*, a linear operator, which transforms the input space on a possibly different output space. If no errors occur, vector spaces are preserved under linear transformations, and if errors do occur, the received vector space U is close to the transmitted vector space V under a distance metric appropriately defined on vector spaces. In other words, if the input spaces (codewords) have a certain minimum distance regarding the number of non-intersecting dimensions, error-correction can be achieved at the decoder provided that the linear network operator is not too rank-deficient and, furthermore, the received space does not contain too many "malicious" dimensions due to error packets.

The authors further observed that low complexity Maximum Rank Distance (MRD) codes, introduced by Gabidulin in [Gab85] can be applied for network coding error detection and correction. The approach introduced by Plass *et al.* [PRV08], originally targeted for crisscross error patterns, can be successfully applied for practical network coding. Some notions of rank codes will be shortly introduced in the following, together with a Berlekamp-Massey algorithm for decoding of rank metric codes, as presented in [PRV08].

1.5.1 Fundamentals of Rank Codes

If $\mathbf{x}$ is a codeword of length n with elements from GF(q^N), where q is a power of a prime, we can consider a bijective mapping from the codeword $\mathbf{x} = (x_1, \ldots, x_n)$ into an $N \times n$ array $\mathbf{A}$.

The *rank metric* over GF(q), or the rank of $\mathbf{x}$ over q, is defined as $\mathbf{A}$, $r(\mathbf{x} \mid q) = r(\mathbf{A} \mid q)$. The rank function $r(\mathbf{A} \mid q)$ is equal to the maximum number of linearly independent rows or columns of $\mathbf{A}$ over GF(q). The rank function can be shown to define a norm ($r(\mathbf{x} \mid q) \geq 0$, $r(\mathbf{x} \mid q) = 0 \Leftrightarrow \mathbf{x} = 0$, $r(\mathbf{x} + \mathbf{y} \mid q) \leq r(\mathbf{x} \mid q) + r(\mathbf{y} \mid q)$, $r(a\mathbf{x} \mid q) = |a| r(\mathbf{x} \mid q)$ is also true if we set $r|a| = 0$ for $a = 0$ and $|a| = 1$ for $a \neq 0$).

If $\mathbf{x}$ and $\mathbf{y}$ are two codewords of length n with elements from GF(q^N), the *rank distance* is defined as: $\mathrm{dist}_r(\mathbf{x}, \mathbf{y}) = r(\mathbf{x} - \mathbf{y}|q)$.

Similar to the minimum Hamming distance, the minimum rank distance of a code C can be determined. For a code C the *minimum rank distance* is given by:

$$d_r = \min \{\mathrm{dist}_r(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in C, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}$$

or, when the code is linear:

$$d_r = \min \{r(\mathbf{x} \mid q) | \mathbf{x} \in C, \mathbf{x} \neq 0\}.$$

Let $C(n,k,d_r)$ be a code of dimension k, length n, and minimum rank distance d_r. It was shown in [Gab85] that there also exists a Singleton-style bound for the rank distance. For every linear code $C(n,k,d_r) \subset GF(q^N)^n$, d_r is upper bounded by: $d_r \leq d_h \leq n-k+1$, where d_h is the minimum Hamming distance. A linear code $C(n,k,d_r)$ is called *Maximum Rank Distance* (MRD) code, if the Singleton-style bound is fulfilled with equality.

In [PRV08], a constructive method for the parity check matrix and the generator matrix of an MRD code is given in the rest of this section.

Construction of MRD Codes

A parity check matrix $\mathbf{H}$ which defines an MRD code and the corresponding generator matrix $\mathbf{G}$ can be written as:

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0^q & h_1^q & \cdots & h_{n-1}^q \\ \vdots & \vdots & \ddots & \vdots \\ h_0^{q^{d-2}} & h_1^{q^{d-2}} & \cdots & h_{n-1}^{q^{d-2}} \end{bmatrix}, \quad G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-1} \\ g_0^q & g_1^q & \cdots & g_{n-1}^q \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{q^{d-2}} & g_1^{q^{d-2}} & \cdots & g_{n-1}^{q^{d-2}} \end{bmatrix}$$

where the elements $h_0, \ldots, h_{n-1} \in GF(q^N)$, and $g_0, \ldots, g_{n-1} \in GF(q^N)$ are linearly independent over GF(q).

The decoding of Rank-Codes with the modified Berlekamp-Massey algorithm is based on linearized polynomials.

A *linearized polynomial* over GF(q^N) is a polynomial of the form $L(x) = \sum_{p=0}^{N(L)} L_p x^{q^p}$, where $L_p \in \mathrm{GF}(q^N)$, and $N(L)$ is the norm of the linearized polynomial (the largest p where $L_p \neq 0$).

Let $\otimes$ be the symbolic product of linearized polynomials defined as:

$$F(x) \otimes G(x) = F(G(x)) = \sum_{p=0}^{j} \sum_{i+l=p} f_i g_l^{q^i} x^{q^p}, \quad \text{with } j = N(F) + N(G).$$

The symbolic product is associative, distributive with respect to ordinary polynomial addition, but non-commutative.

Berlekamp-Massey Algorithm for Decoding Rank-Codes

Let $\mathbf{c}$, $\mathbf{r}$ and $\mathbf{e}$ be the codeword vector, the received vector and the error vector of length n with elements from GF(q^N). The received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$. Let $v = r(\mathbf{e} \mid q)$ be the rank of the error vector $\mathbf{e}$. If $2v < d_r$, the codeword can be correctly decoded:

Syndrome $\mathbf{s}$ is given by:

$$\mathbf{s} = \mathbf{r}\mathbf{H}^\top = (\mathbf{c} + \mathbf{e})\mathbf{H}^\top = \mathbf{e}\mathbf{H}^\top. \tag{1.2}$$

A $v \times n$ matrix $\mathbf{Y}$ of rank v is defined, whose entries are from the base field GF(q).

Then $\mathbf{e}$ can be written in the form:

$$\mathbf{e} = (E_0, E_1, \ldots, E_{v-1})\mathbf{Y}, \tag{1.3}$$

where $E_0, E_1, \ldots, E_{v-1} \in \mathrm{GF}(q^N)$ are linearly independent over GF(q).

Matrix Z is defined as:

$$\mathbf{Z}^\top = \mathbf{Y}\mathbf{H}^\top = \begin{bmatrix} z_0 & z_0^q & \cdots & z_0^{q^{d-2}} \\ z_1 & z_1^q & \cdots & z_1^{q^{d-2}} \\ \vdots & \vdots & \ddots & \vdots \\ z_{v-1} & z_{v-1}^q & \cdots & z_{v-1}^{q^{d-2}} \end{bmatrix} \tag{1.4}$$

where the elements $z_0, z_1, \ldots, z_{v-1} \in \mathrm{GF}(q^N)$ are linearly independent over GF(q). The syndrome equation can be written as: $(S_0, S_1, \ldots, S_{d-2}) = (E_0, E_1, \ldots, E_{v-1})\mathbf{Z}^\top$ or element-wise:

$$S_p = \sum_{j=0}^{v-1} E_j {z_j}^{q^p}, \quad \text{for } p = 0, \ldots, d-2. \tag{1.5}$$

By raising each side of the equation to the power of q^{-p} and after doing the operations in GF(q^N) we obtain:

$$S_p^{q^{-p}} = \sum_{j=0}^{v-1} E_j^{q^{-p}} z_j, \quad \text{for } p = 0, \ldots, d-2. \tag{1.6}$$

This is a system of $d-1$ equations with $2v$ unknowns that are linear in $z_0, z_1, \ldots, z_{v-1}$. The rank v of the error vector is also unknown. It is sufficient to find one solution of the system because every solution of $E_0, E_1, \ldots, E_{v-1}$ and $z_0, z_1, \ldots, z_{v-1}$ results in the same error vector $\mathbf{e}$.

The row error polynomial $\Lambda(x) = \sum_{j=0}^{v} \Lambda_j x^{q^j}$ is a linearized polynomial which has $\Lambda_0 = 1$, and all linear combinations of $E_0, E_1, \ldots, E_{v-1}$ over GF(q) are its roots. The linearized syndrome polynomial can be written as:

$$S(x) = \sum_{j=0}^{d-2} S_j x^{q^j}.$$

The key equation can be defined as:

$$\Lambda(x) \otimes S(x) = F(x) \bmod x^{q^{d-1}}, \tag{1.7}$$

where $F(x)$ is an auxiliary linearized polynomial that has norm $N(F) < v$.

In order to get the row error polynomial $\Lambda(x)$, the following system has to be solved,

with $2v < d$:

$$\sum_{i=0}^{p} \Lambda_i S_{p-1}^{q^i} = 0, \quad \text{for } p = v, \ldots, 2v-1.$$

Subtracting $S_p \Lambda_0$ from both sides and taking into account that $\Lambda_0 = 1$ and $\Lambda_i = 0$ for $i > v$, we obtain:

$$-S_p = \sum_{i=1}^{v} \Lambda_i S_{p-i}^{q^i} = 0, \quad \text{for} p = v, \ldots, 2v-1,$$

which can be written in matrix form as:

$$S \begin{bmatrix} \Lambda_v \\ \Lambda_{v-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_v \\ -S_{v+1} \\ \vdots \\ -S_{2v-1} \end{bmatrix}, \quad \text{where} \begin{bmatrix} S_0^{q^v} & \cdots & S_{v-1}^{q^1} \\ S_1^{q^v} & \cdots & S_v^{q^1} \\ \vdots & \ddots & \vdots \\ S_{v-1}^{q^v} & \cdots & S_{2v-2}^{q^1} \end{bmatrix} \tag{1.8}$$

Matrix S can be shown to be non-singular, so the system of equations has a unique solution. The solution can be found using the modified Berlekamp-Massey algorithm presented in [PRV08]. The overall steps of the decoding procedure can be summarized as follows:

1. Calculate the syndrome with equation 1.2.
2. Solve the key equation 1.8 with the modified Berlekamp-Massey algorithm described in [PRV08] to obtain $\Lambda(x)$.
3. Calculate the linearly independent roots $E_0, E_1, \ldots, E_{v-1}$ of $\Lambda(x)$. This can be done with the Berlekamp-Massey algorithm.
4. Solve the linear system of equations 1.6 for the unknown variables $z_0, z_1, \ldots, z_{v-1}$.
5. Calculate the matrix $\mathbf{Y}$ using equation 1.4.
6. Calculate the error vector $\mathbf{e}$ by equation 1.3 and then the decoded code word $\mathbf{c} = \mathbf{r} - \mathbf{e}$.

The benefits of random linear network coding have made it appealing for several practical applications. However, the effect that corrupted packets can have on such a scheme cannot be neglected. The approach presented in this section shows, on the other hand, that error correcting codes for such a scheme exist, and that they can be easily hardware integrated, similar to the widely used Reed-Solomon codes.

1.6 Conclusion

In this chapter, we presented the concept of network coding, an innovative paradigm alternative to classical routing that allows to maximize the throughput of a network by enabling intermediate nodes to send combinations of the received packets instead of mere copies.

We discussed how this innovation can be beneficial both to non-live applications, such as distributed storage and content distribution, and to delay-constrained applications, such as video streaming and P2P video.

In particular, we reviewed several techniques that combine the features of network coding in general with source coding frameworks, characteristic of multimedia applications over unreliable channels. We showed how these paradigms can be integrated and pointed out the benefits of their interaction, in particular when a joint, cross-layer optimization is performed. We also presented a state-of-the-art technique to deal with errors and erasures that may arise when using network coding.

Chapter 2

Network Coding for Multiple Description Video

Contents

In this chapter we present our network coding solutions for the diffusion of multiple description video in lossy networks, such as wireless ad-hoc networks. First, in Sec. 2.1 we introduce the concept of multiple description coding (MDC), a coding framework which allows improved immunity to losses in unreliable networks without the need for feedback channel or retransmission. In Sec. 2.2 we present our combined MDC and network coding technique for real-time video delivery in ad-hoc network, while our scheduling solution and its validation are provided in Sec. 2.3.

2.1 Multiple description video coding

Multiple description coding (MDC) is a joint source-channel coding technique in which a single media stream is represented by d substreams referred to as *descriptions* that can be delivered over different paths [Goy01]. At the destination, the descriptions can be independently decoded and reproduced, but the quality of the reconstruction improves with the number of descriptions received.

MDC is meant to provide error resilience to the encoded bitstream by allowing any subset of the descriptions to be used to reconstruct the original signal. In this way, common problems like network congestion and packet loss will cause a loss in the quality of the stream, but providing a graceful degradation of the reconstruction rather than an interruption.

In addition to error resiliency, MDC also allows for rate-adaptive streaming, as clients can subscribe to one or more of the descriptions depending on their download capabilities. While in the widely used scalable (layered) coding technique the media stream is partitioned into layers with a base layer and several enhancement layers, each depending on the previous ones in the hierarchical order, in MDC any description will add to the quality of the stream, irrespective of the order in which they are received.

While it was originally developed in the 70's and proposed for speech signals, MDC has been investigated in the field of image and video coding [Goy01], with several solutions proposed. These techniques, some of which will be presented in the following, can be more general, while others depend on the content to be encoded.

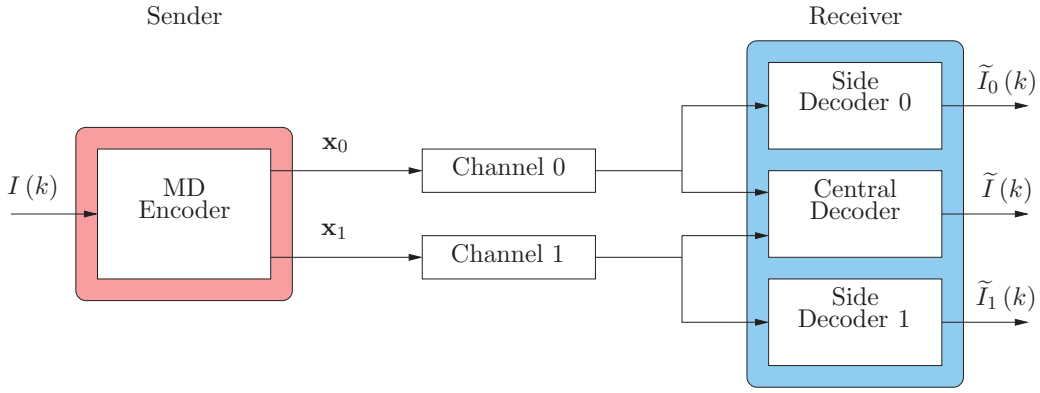
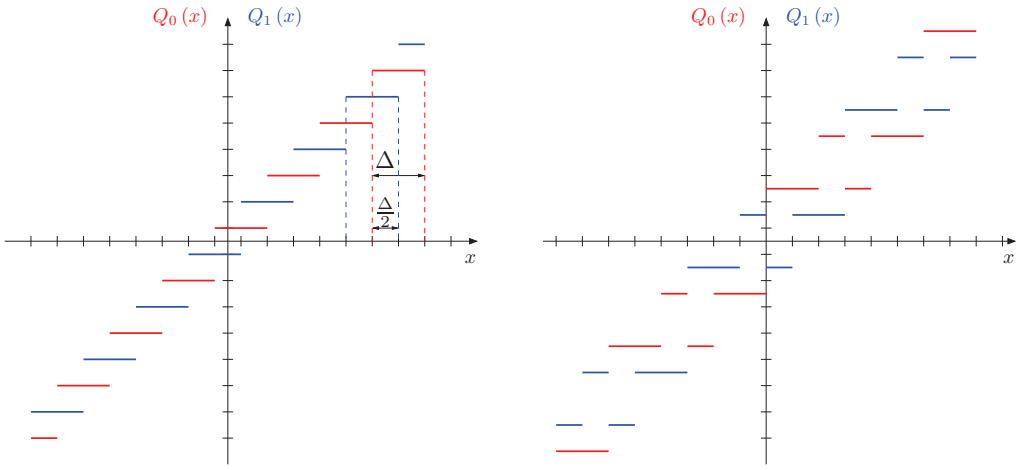


Figure 2.1: Scheme of a two-channels multiple description system. The source signal $I(k)$ is encoded in two *descriptions* $\mathbf{x}_0$ and $\mathbf{x}_1$ sent over two independent lossy channels. When only one description is available, the corresponding *side decoder* generates a reconstructed sequence of acceptable quality, $\widetilde{I}_0(k)$ or $\widetilde{I}_1(k)$. When both descriptions are available the *central decoder* generates a reconstructed sequence of higher quality, $\widetilde{I}(k)$.

The decoding unit used when all descriptions are available is referred to as *central decoder*, while any decoding unit used when a non-empty subset of the descriptions is available is referred to as *side decoder*. A simple two-descriptions transmission system is represented in Fig. 2.1.

One of the first practical approaches to MDC was *Multiple Description Quantization* (MDQ) [Goy01], in which a scalar, vector, or entropy-constrained quantizer is designed to produce a number of descriptions.

In the example depicted in Fig. 2.2(a), the multiple descriptions are obtained with two uniform quantizers of step Δ, the second one being offset by half a quantization interval with respect to the first one. If one description is lost, the source signal is reconstructed from samples quantized with a step of Δ; if both descriptions are received, the resulting quantization step is $\Delta/2$.



(a) Offset quantiser (b) Non-convex quantizer

Figure 2.2: Two examples of multiple description scalar uniform quantizer.

In this scheme, if the side quantizers have b-bit resolution, the central decoder has approximately $(b+1)$-bit resolution. In other words, when no losses occur, the system is using $2b$ bits in order to have a resolution of $b+1$ bits, a very high redundancy that discourages the use of this technique unless the channel loss rates are very high and side reconstructions are very important. In order to overcome this drawback several techniques with lower redundancy have been proposed, such as the non-convex MD-quantizer [Reu80], exemplified in Fig. 2.2(b).

A theoretical framework for designing MD scalar quantizers with fixed-rate quantization was later independently proposed by Vaishampayan *et al.* [Vai93], subsequently extended to entropy-constrained quantization [VD94].

In MDQ the multiple descriptions are generated by the quantizer. Quantization occurs in any kind of lossy coding, independently from the nature of the content (audio, image, video *etc.*). Schemes based on multiple description scalar quantizers and the H.264/AVC standard have been proposed for reliable real-time video applications [GGP01, CCM08]

The subject of MDC has been extensively researched in the past 20 years with many solutions proposed in the context of images or video [WRL05]. A simple solution to obtain two descriptions is *spatial splitting*, which consists in partitioning each individual frame of

the video sequence [FFLT05]. Spatial splitting is based on the spatial correlation among neighboring samples, so a separation of the even and odd rows or columns in the frames can yield two balanced descriptions.

Another simple method for building two or more descriptions is *temporal splitting*. It consists in the separation of the odd and even frames, that are to be encoded independently [Apo01]. When both descriptions are received, the two reconstructed sub-streams are merged to create the central reconstruction. The merging technique can be a simple interleaving of the frames of the two sub-streams, or a combination of received and interpolated sequence.

When one description is missing, its samples can be approximated by temporally interpolating the other sub-stream. The interpolation technique can be as easy as sample-wise sample-and-hold interpolation, which is equivalent to reducing the frame rate of the reproduction, or a more sophisticated technique such as *motion compensated temporal interpolation* [GCPP10, GPCPP11]. This technique has been shown to provide high compression ratios, especially in regular motion video (such as video conferencing [FFLT05, WRL05]), it is easy to implement, and the descriptions can be encoded with standard encoders, such as H.264. This is also important as, it does not exist to the day a standard explicitly addressing multiple description video coding.

In the spatial or temporal splitting techniques, the natural correlation between symbols in the source signal is exploited for reconstruction, *e.g.*, odd samples can be predicted from even samples, and vice-versa. Thus, the degree of correlation among the descriptions depends only on the statistics of the input signal.

A considerably different approach is *MD transform coding*, in which a linear transform is actively designed to control the degree of correlation between the descriptions of the source signal. Transform coding represents one of the most performing solutions for MDC [GKAV98, GK01, WOVR01], as it provides good energy compaction properties, resilience to additive noise and quantization, and freedom to capture important signal characteristics. The correlation that remains after signal transformation can mitigate the effect of losses, as the lost elements can be estimated based on the received ones.

In conventional (single description) video coding, redundancy among samples is reduced via transform coding, so that the transform coefficients are less correlated and more compact. In the MD case, the quantized versions of the transform coefficients are to be sent over different channels. If one of the channel fails and one description is lost, since the coefficient are poorly correlated, there would be no way to estimate the other.

To prevent this, a known correlation can be introduced in order to allow an estimation of a missing description. For example, a signal $\mathbf{x}$, consisting of two independent Gaussian variables x_1 and x_2, is transformed into $\mathbf{y}$, whose components are $y_1 = x_1 + x_2$ and $y_2 = x_2 - x_1$, which has been shown to be optimal for independent Gaussian sources [WOR97]. This transformation is such that the statistical dependencies between the components of $\mathbf{y}$ allow from any one of them to estimate the original two components of $\mathbf{x}$ to a certain

accuracy, and the total estimation error for either component is the same.

In practice, the decorrelating and a pair-wise correlating transform is implemented as a single linear transform such that coefficients intended to the same description are internally decorrelated, and coefficients intended to different descriptions are correlated with each other [WOVR01].

Since it is possible to obtain an acceptable reconstruction of all coefficients given one description, the RD performance of the side decoder is improved, but the performance of the central decoder is degraded.

This method has been introduced for two descriptions in the context of image coding [WOR97], and subsequently extended to more general mappings to produce an arbitrary number of descriptions [GK98, GK01].

It is worth noticing that, while in SDC quantization is performed after the transformation, quantizing before applying a correlating transform has been shown to give the best performance in MDC schemes [OWVR97, WOVR01].

Another possibility to generate correlated representations is to project the signal onto an over-complete signal dictionary. For discrete signals, since the number of output coefficients will be larger than the number of input signal samples, the different subsets of the coefficients can be included in different descriptions and sent over independent channels. Redundant transformation of the input signal was also proposed as a way to achieve multiple description coding in the context of transmission with an unpredictable number of losses [GVK98, GVT98, GKV99].

Other methods based on the redundant wavelet transform present also the advantage of allowing scalability. A survey on redundant wavelet schemes for multiple description coding of video sequences can be found in [TPPP07]. More recently, the problem of optimal rate allocation for redundant transform MD schemes has been addressed, adapting the quantization of the transform coefficients to the importance of the basis functions, the redundancy in the representation, and the expected loss probability on the channel [RF07].

2.2 A framework for joint Multiple Description Coding and Network Coding

In this section, we model the diffusion of a content, encoded in D descriptions, on a Mobile Ad-hoc Network (MANET), from a single sender to a multitude of receivers. We assume that each node of the network will contribute to the diffusion, but it will be able to sustain an up-link bitrate sufficient for only one description.

Several works [IKLAA11, RW10] have already proven that MDC video multicast can benefit from the use of network coding. For instance, Ramasubramonian *et al.* proposed a centralized method in which the server has complete information about the number of descriptions obtained by each user. However, this method cannot be applied in the case

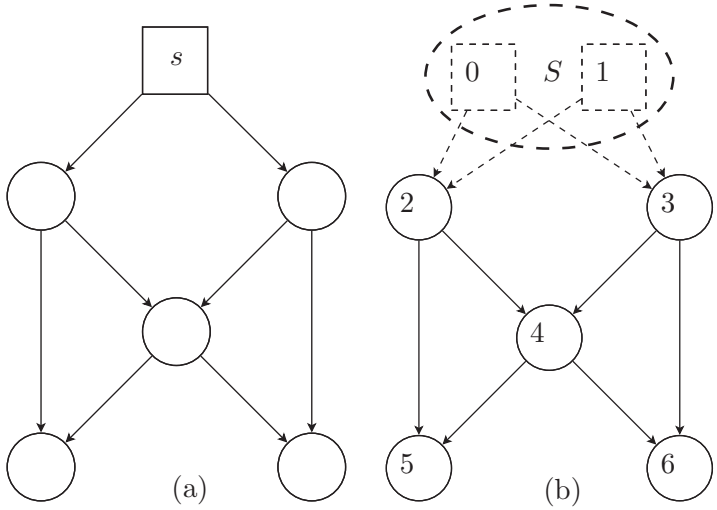


Figure 2.3: (a) Butterfly wireless network.
(b) Equivalent network. The source s is modeled with two virtual sources 0 and 1 to have uniformly unit capacity of channels. The remaining nodes i are labeled in non-decreasing height order.

of multi-hop wireless networks with node mobility.

We start our discussion from the model proposed by Chou *et al.* [CWJ03] for Practical Network Coding, where the communication network is represented by a directed acyclic graph having unit capacity and denoted by $G = (V, E)$. Let $s \in V$ be the sender, or *source node* and $T \subset V$ the set of receivers, or *peers*. In our scenario for video dissemination we assume that the set of receivers includes any node except the source, $T = V - \{s\}$.

We assume the video is MDC-encoded in D descriptions. At each sending opportunity, the source has to broadcast a video frame x, encoded in packets $x_0, x_1, \ldots, x_{D-1}$ (one per description), to each peer in the network. In the standard model, each transmitted symbol is associated to the edge (or *channel*) it is carried over. We define $y(e_{ij})$ the symbol carried over channel $e_{ij} \in E$. Since the network we want to model is a broadcast network, we need to introduce a further constraint. Namely, we impose that each node n_i transmits the same symbol, denoted y_i, over all its outgoing channels.

Once all symbols y_i are assigned, it is possible to revert to the standard model by setting $y(e_{ij}) = y_i$, $\forall e_{ij} \in E$.

Imposing that for all nodes the same symbol is sent on all unitary capacity channels raises a problem in our model: the source would be constrained to send packets from just one out of D descriptions. To avoid this problem, we can model the video source as a set S of D virtual sources, each having a copy of the outgoing channels of s and emitting exactly one description of x over all its channels, thus maintaining uniformity in the unit capacity of channels:

$$y(s_d) = x_d, \quad \forall d \in \{0, 1, \ldots, D-1\}\,.$$

As it can be seen in Figure 2.3, by our definition the set V does not contain the "original" source but instead the set S. There are thus $N = |V|$ nodes, D virtual sources and $N - D$ peers in our model.

Symbols emitted by a node $i \in T$ must be linear combinations of the symbols carried over the channels entering i. Let us define $\mathcal{N}_i$ the set of nodes j such that a channel exists from j to i. The symbol emitted by i will be in the form:

$$y_i = \sum_{j \in \mathcal{N}_i} m_i(j) y_j, \qquad \forall i \in T. \tag{2.1}$$

The *local encoding vector* of node i, $\mathbf{m}(i) = [m_i(j)]_{j \in \mathcal{N}_i}$, represents the encoding function of node i along all its outgoing channels $e_{ik} \in E$. In the case of the source, which does not have any entering channels, its emitted symbol is assigned beforehand.

Let us define the height of a node h_i as the length of the *longest* finite path in G from any node in S to node i; this induces a partial order on set V, well defined, as G is acyclic. We label the nodes with indices $0, 1, \ldots, N-1$ such that $i < j \implies h_i \leq h_j$ (see Fig. 2.3). Accordingly, the nodes in the source set S are labeled $0, 1, \ldots, (D-1)$.

This order is consistent with the propagation of packets outgoing from the source. If we define an *encoding matrix* $\mathbf{M}$ as follows:

$$M_{ij} = \begin{cases} m_i(j) & \text{if } e_{ji} \in E, \\ 0 & \text{otherwise,} \end{cases}$$

we can rewrite equation (2.1) as:

$$y(i) = \sum_{j<i} M_{ij} y(j), \qquad \forall i \in \{D, D+1, \ldots, N-1\},$$

while for the virtual sources we impose

$$y(i) = x_d, \qquad \forall d \in \{0, 1, \ldots, D-1\}.$$

We are interested in the number of packets a node is able to decode. Let us define $\mathbf{x}$ as a column vector that has for components all the descriptions of frame x, that is:

$$\mathbf{x} = [x_d]_{d \in \{0,1,\ldots,D-1\}}.$$

Since any emitted symbol $y(i)$ is a linear combination of packets from all the descriptions, there exists a row *weight vector* $\mathbf{w}(i)$ with D components such that $y(i) = \mathbf{w}(i) \cdot \mathbf{x}$. The vector $\mathbf{w}(i)$ for nodes in S has only one non-zero component, corresponding to the

description emitted,

$$\mathbf{w}(i) = [\delta_{\mathrm{d,i}}], \qquad d = \{0, 1, \ldots, D-1\}, \qquad i = \{0, 1, \ldots, D-1\},$$

while for nodes in T it can be inferred from matrix $\mathbf{M}$:

$$\mathbf{w}(i) = \sum_{j<i} M_{ij}\mathbf{w}(j), \qquad \forall i \in \{D, D+1, \ldots, N-1\}.$$

Nodes transmit their weight vector along with their emitted symbol; the nodes that receive this information interpret it as a linear equation in the form:

$$y(j) = w_0(j)x_0 + w_1(j)x_1 + \cdots + w_{D-1}(j)x_{D-1}.$$

By collecting symbols and weight vectors on its entering channels, a node i is able to construct a system of linear equations:

$$\mathbf{W}(i)\mathbf{x} = \mathbf{y}(i),$$

where $\mathbf{W}(i)$ is a *weight matrix* obtained by horizontal concatenation of vectors $\mathbf{w}(j)$, and $\mathbf{y}(i)$ is a column vector with components equal to $y(j)$, for all j such that $e_{ji} \in E$.

A node i is able to perform *central decoding*, *i.e.*, to decode all D descriptions, if and only if $\mathbf{rank}(\mathbf{W}(i)) = D$. However, the rank is not a reliable tool to estimate the number of descriptions used in *side decoding*, *i.e.*, decoding only a subset of the D descriptions. For instance, a node i having $\mathbf{rank}(\mathbf{W}(i)) = 1$ could be receiving an equation from node j in the form $y(j) = w_0(j)x_0$, which is trivial and allows the decoding of x_0. But it could also be receiving $y(j) = w_0(j)x_0 + w_1(j)x_1$, which is impossible to solve without further information.

Let us assume we have an operator $\mathbf{dec}(\mathbf{W}(i))$ able to infer how many descriptions a node i will be able to decode, given its weight matrix $\mathbf{W}(i)$. This operator can be easily implemented in practice by counting the number of trivial equations. We also define a *value* operator $\varphi(\cdot)$ that associates a quality metric to a number of decoded descriptions $\mathbf{dec}(\mathbf{W}(i))$. The choice of the quality metric will depend of course on the requirements of the application. If we use, for instance, the expected PSNR, $\varphi(\cdot)$ should reflect the fact that the PSNR gap between a node not receiving any description and a node receiving just one is bigger than the gap between a node receiving one description and a node receiving two, *i.e.*, $\varphi(1) - \varphi(0) \geq \varphi(2) - \varphi(1)$.

Given this model, our approach is the following. At each sending opportunity, each node i inspects the state of the buffers of its neighbors, then it chooses an optimal weight

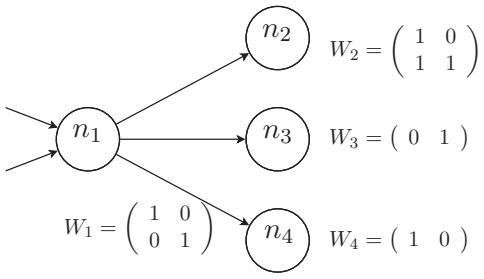


Figure 2.4: Example of optimization of the weight vector of a node n_1.

vector $\mathbf{w}^*(i)$ as:

$$\mathbf{w}^*(i) = \arg\max_{\mathbf{w}\in\mathcal{W}_i} \left\{ J(\mathbf{w}) = \sum_{j\in\mathcal{N}_i} \varphi\left(\mathbf{dec}(\mathbf{W}(j))\right) \right\}, \tag{2.2}$$

where $\mathcal{W}_i$ is the set of weighting vectors available to i, and $\mathcal{N}_i$ is the set of indices of the neighbors of n_i. Notice that $\mathcal{W}_i$ is restricted by the fact that a node i can only choose to send one of the packets it received or a combination thereof.

The optimization of the emitted symbol is independent from the decoding capability, *i.e.*, even if a node is unable to decode any description (*e.g.*, if it is receiving just a combination $w_0x_0 + w_1x_1$), it could choose to forward a combination it received if that would benefit its neighbors.

In Fig. 2.4 we present an example of optimization performed by a node n_1. Here, $J([1,0]) = \varphi(2) + \varphi(2) + \varphi(1)$, as only the rank of W_3 is affected by the combination, $J([0,1]) = \varphi(2) + \varphi(1) + \varphi(2)$ as only the rank of W_4 is affected, and $J([1,1]) = \varphi(2) + \varphi(2) + \varphi(2)$ as both the rank of W_3 and W_4 are affected. Since $\varphi(2) > \varphi(1)$, the optimal choice is $w_1^* = [1,1]$.

There are two main challenges that need to be dealt with in order to use this approach: firstly, a mobile ad-hoc network is hardly a DAG; secondly, the nodes need to inspect the buffer state of their neighbors in order to solve the optimization problem (2.2).

We therefore apply the algorithm not directly on the ad-hoc network, but rather on an *overlay network* built on top of it. We generate a DAG overlay using the ABCD protocol [GC11, GCPP12], a cross-layer protocol for content delivery over MANETs.

ABCD forms an overlay consisting in the superposition of D different diffusion trees, one per each description, and is therefore acyclic. From the transmission point of view, ABCD has a modified 802.11 MAC layer, which implements a reservation mechanism that ensures reliable broadcast, at the same time reducing the collision probability. Such a strategy has been shown to allow a better performance than the standard 802.11 when considering the trade-off between the rate and the diffusion area in multi-hop wireless networks [EFLBS07]. The protocol is able to quickly adapt to topology changes (movement,

nodes' departure or arrival) with a small number or exchanged messages.

Thus the control messages sent in order to build and maintain the overlay can be used to propagate information about the local buffer $W(i)$ from node i to its neighbors. The underlying ABCD protocol also ensures that the nodes have an up-to-date view of the topology, even in presence of node mobility and churn, and minimizes the protocol overhead.

2.2.1 Experimental results

In the following, we present the results of the proposed technique and compare them with the result achievable via the Practical NC scheme proposed by Chou *et al.* [CWJ03].

To generate the DAG, we randomly construct a MANET consisting of 100 nodes in a $100 \times 100\,\mathrm{m}^2$ playground. The nodes have a nominal transmission range of 25 m, then the ABCD protocol [GC11] is run in order to form a directed acyclic overlay. When the source starts broadcasting the stream, the proposed coding strategy is applied at each node. Observing which packets have been decoded, we compute average PSNR observed by the various users.

In order to generate the video content we encoded the "Foreman" video sequence (CIF, 30 fps, 288 frames) using a recently proposed MD coder [GPCPP11]. The stream is encoded in two descriptions, $D = 2$, balanced in terms of rate-distortion properties. Since the technique combines only frames on the same prediction level, the two frames have similar size and we can just zero-pad the smallest to match the sizes before combining them.

The test is repeated 100 times, with different initial positions of the nodes, in order to take into account the variability of the network topology. Several tests have been performed with other video contents, with similar results.

The results for the average PSNR obtained with our method and PNC can be seen in Fig. 2.5. The theoretical bound reported for reference was obtained by exhaustive exploration of the solutions.

For the PNC implementation we assumed that the coding window cannot be set along the time axis, *i.e.*, we do not mix packets with different due-dates, in order to avoid decoding delay, crucial in real-time applications. Therefore, combination of packets can only occur along the descriptions axis, *i.e.*, we mix packets from different descriptions, but having the same due-date. This implies that the length of the coding window equals the number D of descriptions, which is 2 in our test. The coding coefficients are chosen randomly in the finite field of size 256 (*i.e.*, GF(256)), which has been shown [CWJ03] to give a low probability of building duplicate packets. In order to reduce the overload on the network, in the PNC scheme used for reference we employ probabilistic flooding, with each node having a forwarding factor of 75 %, chosen to also maximize the average PSNR.

We notice that the proposed technique performs on average about 2 dB better than PNC. Moreover in Fig. 2.6, we report the PSNR cumulative distribution functions for the

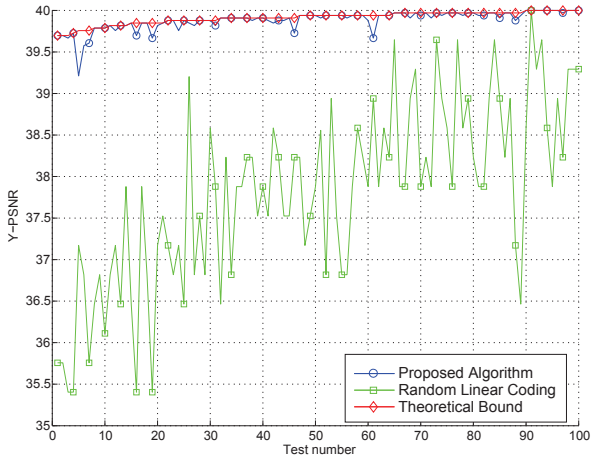


Figure 2.5: Comparison between the reference and the proposed technique, for video sequence "foreman", CIF, 30 fps, 1.8 Mbps. The theoretical bound reported for reference was obtained by exhaustive exploration of the solutions.

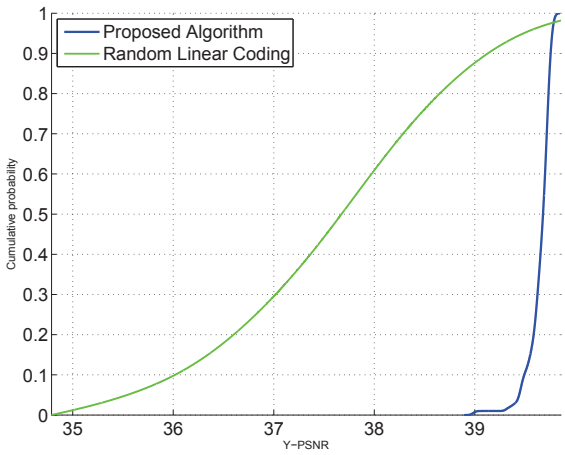


Figure 2.6: Comparison of PSNR CDFs for the same conditions as Fig. 2.5.

two techniques. We observe that while in the reference technique the distribution of PSNR is very widespread, using our technique almost all nodes achieve very close, and very high qualities. We ascribe this result to the fact that the efficiency of PNC is considerably affected by the length of the coding window (*i.e.*, the size of the generation).

To achieve these qualities, using the reference technique all nodes collectively injected into the network 4.6×10^3 packets per second on the average, where as with our technique, only 1.2×10^3 packets are sent. We also noticed that from one experiment to another, these values do not vary much.

We conclude that in this particular scenario, with stringent constraints on delay, an optimization technique outperforms a technique based on random coefficients, providing a better video quality to the end users.

2.3 Scheduling for streaming of Multiple Description Video over wireless networks

While in our previously presented method the routing and network-coding of multiple description packets benefits from a transmission overlay that supports the exchange of information among nodes, in our second contribution the optimization is performed without any input from the receivers.

In this section, we detail the proposed framework, whose objective is to provide a novel transmission strategy for lossy wireless networks able to guarantee a good trade-off between resiliency to losses and timely delivery.

In our scenario, a set of M uncoordinated sources transmit the same encoded video sequence to a single receiver. This scenario could model, for instance, a single hop of a multi-hop transmission.

We propose to jointly use *network coding with an expanding window* (EWNC) and video MDC, which we expect to provide loss resiliency to the video stream without affecting the delay. EWNC was initially proposed [VS10, VS12] to provide unequal error protection (UEP) to random linear coding (RLC) strategies, for the transmission of source messages containing packets of different importance over lossy links. The authors have provided an exact decoding probability analysis for the different layers of the source data in random linear network coding designs. EWNC was lately also proven effective in MDC streaming over wireless networks [NSV12].

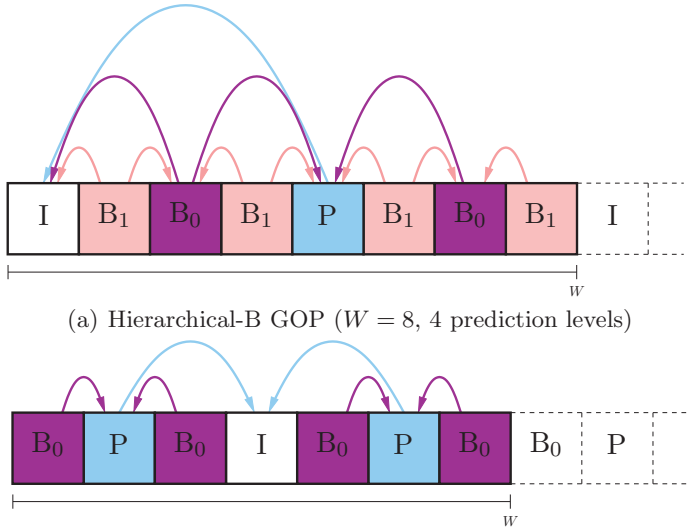
The key idea of EWNC used in our method is to increase the size of the coding window (*i.e.*, the set of packets in the generation that may appear in combination vectors) for each new packet. If in the buffer of the receivers the received coding vectors are kept in row echelon form, using Gaussian elimination, this method provides earlier decodability of packets if no losses occur.

However, the efficiency of EWNC highly depends on the order in which the packets are included in the coding window. The original EWNC method was proposed for layered video coding, therefore the priority of the packets was naturally imposed by the dependencies among layers. Such a strategy is unfeasible in our scenario, as we deal with multiple uncoordinated senders sharing a broadcast medium, and if they all were to choose the same order of packets (*i.e.*, the one imposed by the layered structure), at any given sending opportunity they would send *non-innovative* combinations.

In general, if a prioritization is optimal, it is also unique, and thus all the senders would always transmit dependent combinations, defeating the purpose of using NC. However, there exist frames with very similar RD-properties, hence we can generate a variety of scheduling slightly suboptimal, but with performances very close to the optimum.

For instance, the GOP structure of a video coding technique (such as H.264/AVC or HEVC) leaves a certain degree of freedom in the scheduling, as frames on the same prediction level can be sent in any order (two examples of GOP structures are shown in

Fig. 2.7), but this may not be enough to provide a sufficient number of different schedules for the different senders.



(a) Hierarchical-B GOP ($W = 8$, 4 prediction levels)

(b) Tree GOP ($W = 7$, 3 prediction levels)

Figure 2.7: Two possible GOP structures in H.264/AVC. Arrows indicate reference frames for prediction. Frames on the same prediction level can be sent in any order.

Using an MDC technique, it is possible to have multiple senders transmitting packets that refer to the same instant, but different nonetheless. Furthermore, corresponding packets of different descriptions are mutually refinable, therefore a node being served by multiple senders will perceive an enhanced video quality.

Using MDC, the pool of frame candidates for inclusion in the coding window is a bi-dimensional *multiple description GOP* (MD-GOP), *i.e.*, a rectangular buffer of size $N \times W$, where N is the number of descriptions and W is the GOP size of each description. An example of MD-GOP is depicted in Fig. 2.8, for $N = 4$ descriptions and a GOP structure of each description as the one in Fig. 2.7(a), *i.e.*, Hierarchical-B. Notice that in the buffer the frames are not ordered by their play-out date, but in the encoding order, so that frame dependencies are respected.

The task of the scheduler is to provide an order in which the frames in the MD-GOP are included in the coding window. Since wireless networks are affected by churn and mobility and the video stream can be interrupted at any moment, it is desirable that any new combination maximizes the marginal benefit in terms of RD properties. In other words, at each step, we want the scheduling algorithm to select the frame that optimizes an RD criterion for insertion in the coding window.

However, the corresponding frames of different descriptions might have differences in their RD properties, which would still lead to a unique optimal policy of inclusion in the coding window.

In order to overcome this problem, we propose a *clustering* of the video frames. The

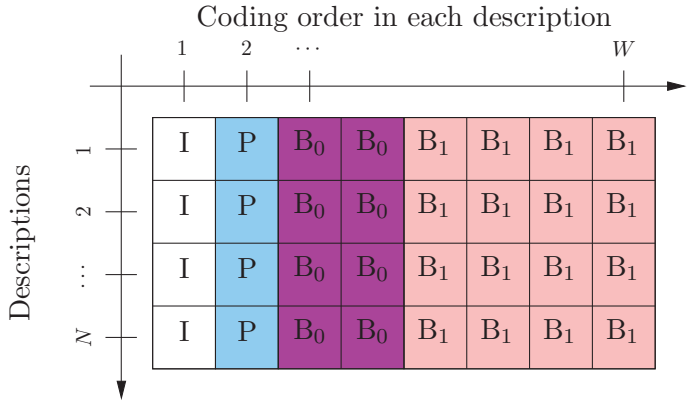


Figure 2.8: MD-GOP for $N = 4$ descriptions and $W = 8$ frames in Hierarchical B-frame GOP. Frames are ordered by prediction level.

clustering is a classification of the frames that takes place at video source, after the video encoding and before scheduling for transmission. Its purpose here is to improve diversity by letting nodes transmitting, at each sending opportunity, a random frame within an optimal cluster.

Clusters are decided once at the encoder, where rate and distortion are known with negligible computational overhead, with frames in the same prediction level. Clustering can be performed in several ways. For instance, a coarse but simple scheme is to assign all the frames on the same prediction level to a single cluster. This scheme is independent from the actual RD properties of the sequence and can be easily implemented; nevertheless, it can be quite efficient if the descriptions are actually frame-by-frame balanced. If the corresponding frames of different descriptions have slightly unbalanced properties, then a more sophisticated scheme can be employed, for instance a gradient descent method that sets the thresholds of each cluster by minimizing the variances of rate and distortion within the cluster. The number of clusters can either be fixed in advanced or determined by the algorithm itself using a Lagrangian constraint on the number of clusters.

The average rate and distortion of the cluster $R(c)$ and $D(c)$ are then computed, possibly quantized, and added as a header to each frame in the cluster. At each sending opportunity, among the clusters whose prediction level is compatible with the scheduling so far ($\mathcal{C}$), each sender chooses the cluster c that minimizes the cost function $J^* = \min_{c \in \mathcal{C}} \{J(c) = R(c) + \lambda D(c)\}$. Within this cluster, each sender *randomly* chooses one frame and schedules it for transmission. This frame is added to the encoding window, increasing its size by one. The size of the coding window is reset to one with the new GOP. A summary of the operation performed by the nodes is reported in Algorithm 1.

An example of frame clustering is presented in Figures 2.9 and 2.10. There, the I-frames of the 4 descriptions have roughly the same RD properties and are therefore assigned to a single cluster. On the P-frames, on the other hand, descriptions 1 and 2 have similar RD properties between them, but different from descriptions 3 and 4, which are in turn close

Algorithm 1 Algorithm used by the nodes to include the frames in the coding window.

procedure SelectFrame
  $G \leftarrow N \times W$; ▷ Size of the generation.
  for all MD-GOPs **do**
    set size of coding window to zero;
    for $i \leftarrow 1$ **to** G **do**
      $\mathcal{C} \leftarrow$ {Pool of eligible clusters};
      $c^* \leftarrow \arg\min_{c \in \mathcal{C}} \{J(c) = R(c) + \lambda D(c)\}$;
      $f^* \leftarrow$ a random frame in cluster c^*;
      increase size of coding window by one;
      include f^* in coding window;
    end for
  end for
end procedure

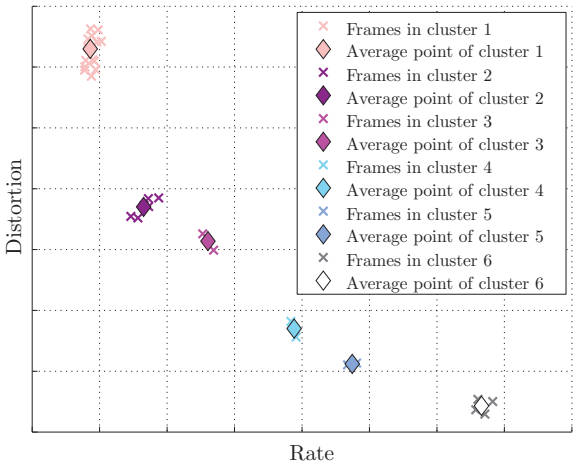


Figure 2.9: Example of video frames clustering for RDO-scheduling. Frames with similar operating points are assigned to the same cluster. The RDO-scheduling will consider each frame as having the average operating point of its cluster.

to each other. In this case, two clusters are created containing the frames with similar properties. The same holds true for the B_0-frames, where descriptions 1, 2 and 3 have been clustered together, while description 4 was assigned to another cluster. Finally, all B_1 frames of all descriptions give similar contributions to distortion and have been assigned to a single cluster.

Large clusters increase the diversity of the scheduling among senders, thus reducing non-innovative packets. However, if clusters are chosen too large, the scheduler will randomly choose among frames with very different values of the objective function, resulting in a sub-optimal performance.

Ideally, the size of the clusters should be chosen according to the expected number of senders that are going to transmit at the same time, which can be roughly estimated with the node density of the network.

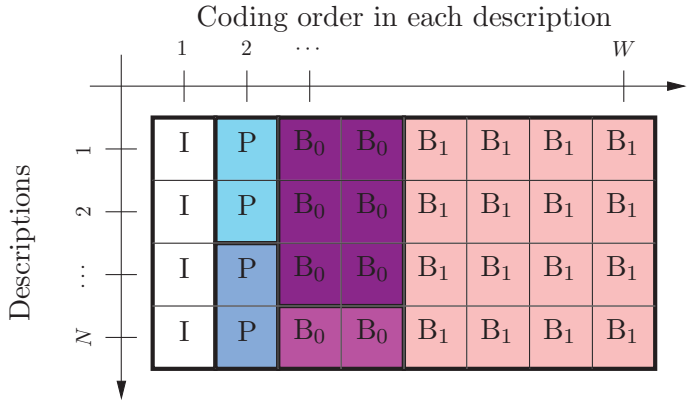


Figure 2.10: Example of MD-GOP clustering. Frames marked by the same color are in the same cluster and share similar RD properties.

An example of two different scheduling orders compatible with the clustering of Figure 2.10 is presented in Figure 2.11. For the sake of clarity, only the scheduling for the first 16 packets is presented. We can observe that, if only a subset of a cluster is chosen, the two schedulers choose different frames within it. If the whole cluster is chosen, then the frames still differ in the order they are included in the coding window.

2.3.1 Experimental results

In the following, we present the results of the proposed technique and compare them with the results achievable with EWNC applied to an SD-coded stream and EWNC applied on an MD-coded stream, but ordered using a trivial schedule. For SDC, the trivial strategy consists in including the frames in coding order, *i.e.*, by prediction level and, within frames on the same level, play-out order. For MDC, we assume again that frames are included in coding order and, within frames with the same encoding time (*i.e.*, corresponding frames of independently encoded descriptions), the descriptions are selected in a fixed order, randomly chosen by each sender independently.

To encode the video sequences, we chose to use four descriptions *Polyphase Down-sampling Multiple Descriptions* (PDMD) [FFLT05, CFL00], a technique where N sub-streams are generated by splitting the original sequence via polyphase down-sampling along rows and columns by a factor of two in the spatial domain. To generate the descriptions, each sub-stream is independently encoded using an H.264/AVC reference encoder JM [Süh11], version 17.0, using the same QP for all descriptions. The encoding algorithm uses the closed-GOP structure presented in Fig. 2.7(a). A closed-GOP was preferred in order to reduce error propagation in case of losses.

The RD properties of each frame are exactly measured. Clustering is performed based on prediction level. The average rate and distortion for the frames in each cluster are computed, quantized on eight bits each, and sent along with the video data.

At the decoder side, all the descriptions are independently decoded in order to obtain

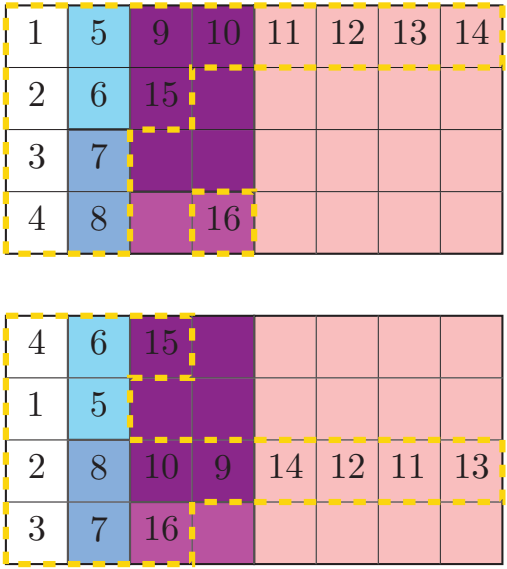


Figure 2.11: Two possible schedules (first 16 packets). The numbers indicate the order in which the frame is included in the coding window. The dashed border identifies which frames have been selected for inclusion in the coding window at the 16-th packet.

the N sub-streams, which the receiver interleaves to reconstruct the central sequence. When some descriptions are lost, the receiver interpolates the missing pixels from the available substreams to obtain a good low-resolution frame (side decoding). When none of the descriptions is available, the loss is concealed using the temporally closest decoded frame.

In order to compare the performance of the method under a variety of inputs, we selected a set of 10 QPs (in Tab. 2.1) and 8 video sequences (in Tab. 2.2) with CIF spatial resolution at 30 frames per second.

High Bitrate	16	19		
Medium Bitrate	22	25	28	31
Low Bitrate	33	36	39	42

Table 2.1: QPs used in encoding the video sequences.

akiyo	hall	foreman	city
coastguard	football	stefan	bus

Table 2.2: Video sequences used in simulations.

The transmission scenario we simulate is depicted in Fig. 2.12. In this scenario, M sources S_m, $m = 1, \ldots, M$, intend to transmit the same video sequence, $I(k)$, $k = 1, \ldots, K$, to a single receiver R.

In order to allow a clear evaluation of our technique, a discrete-time transmission model is assumed: the time is segmented in *transmission rounds* wherein each source S_m sends exactly one packet from a predetermined transmission buffer TX_m. Each channel

C_m between transmission buffer TX_m and the receiver buffer RX is in general lossy, with independent uniform packet loss probability p_m; the transmissions on different channels do not interfere with each other. At the end of each round, the receiver decodes all the frames available in its buffer RX, generating a reconstructed sequence $\widetilde{I}(k)$.

This simple scenario is well suited to model a wireless ad-hoc network where a channel reservation mechanism is enforced [GC11], which provides both discrete-time transmission and channel isolation.

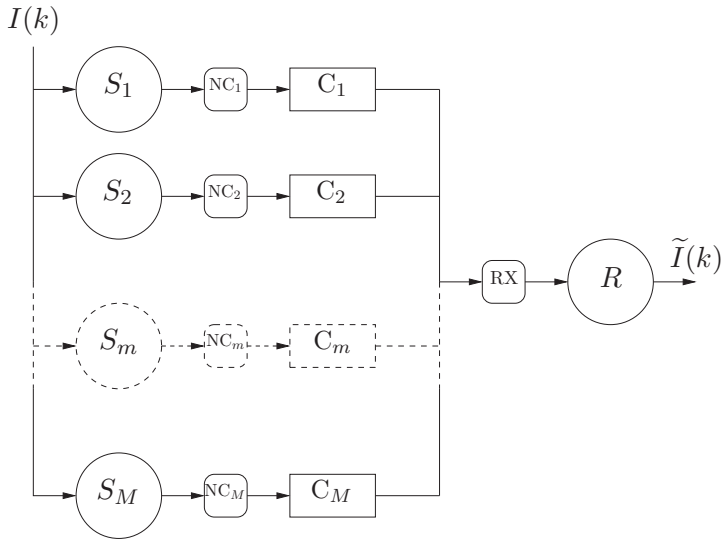
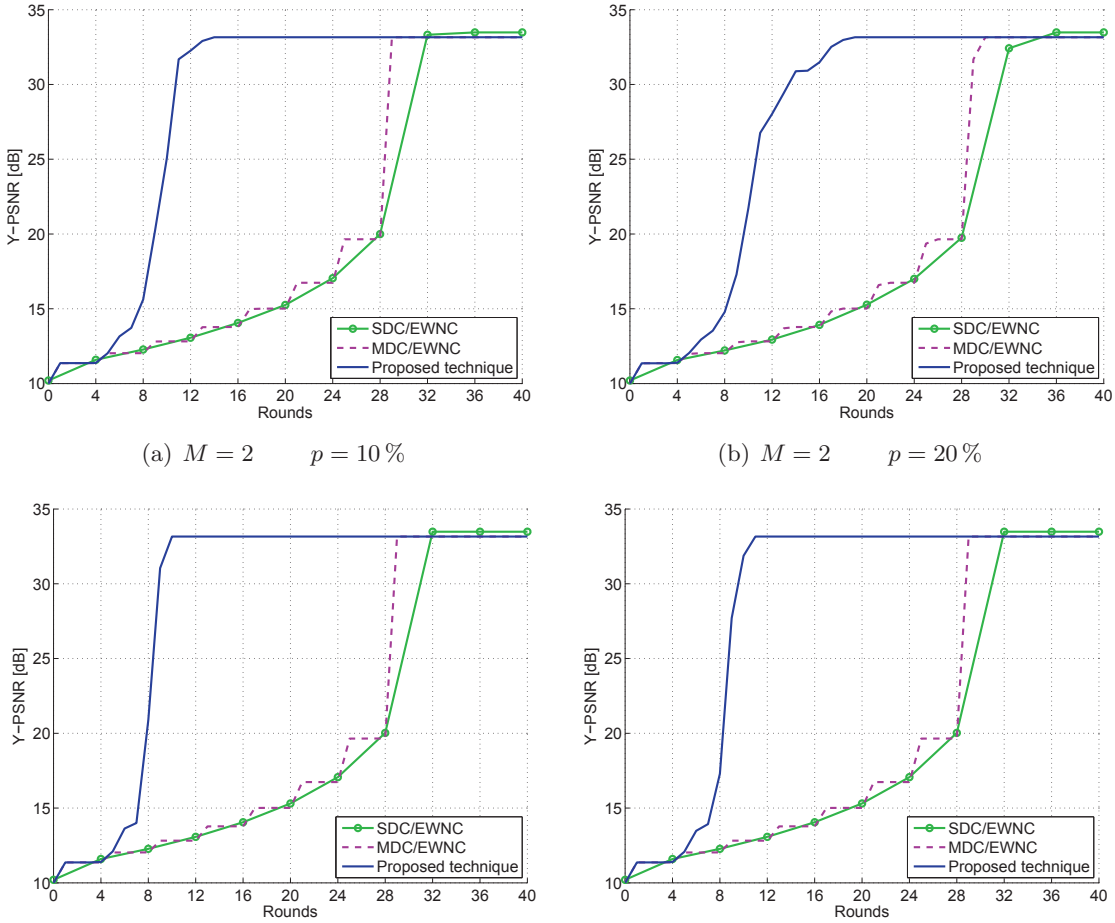


Figure 2.12: Simulated scenario. $I(k)$ and $\widetilde{I}(k)$ are respectively the original and reconstructed frames, S_m, $m = 1, \ldots, M$ are the senders, NC_m the network coding modules, C_m the channels, RX is the receiver R's buffer.

In our simulations, the proposed approach has proven to be able to deliver an acceptable video quality to the receiver in a shorter number of rounds than the reference techniques. As an example, in Fig. 2.13, we report a comparison with the reference techniques under a few different simulation conditions.

We observe that, thanks to the variety in the scheduling, our technique is able to reduce the number of linearly dependent coding vectors, and is therefore able to provide a better video quality (in terms of PSNR) in fewer rounds. It should be noted that the final value of the PSNR for the SD-based technique is slightly higher (about 0.5 dB) than that of both MD-based ones, which is a direct consequence of the inherent redundancy among the descriptions of the MD encoding. However, this happens after a long enough time (*i.e.*, about 30 rounds), during which MDC/NC has already achieved its final PSNR. We can also observe that the performance of the method benefits from a higher number of sources, whereas it is of course negatively affected by the loss rate.



(a) $M = 2$ $p = 10\,\%$ (b) $M = 2$ $p = 20\,\%$

(c) $M = 4$ $p = 10\,\%$ (d) $M = 4$ $p = 20\,\%$

Figure 2.13: Comparison of the average PSNR of the decoded sequences, for M sources and packet loss probability $p_n = p$ for all channels.

2.4 Conclusions

In this chapter we presented two novel techniques of video diffusion over ad-hoc networks based on the joint use of network coding and multiple descriptions coding.

For the first contribution, we formulated the problem of broadcasting a video stream encoded in multiple descriptions on an ad-hoc network in terms of finding an optimal set of combination coefficients. Then, we introduced an objective function that takes into account the effect that decoding a given number of descriptions has on the total distortion. This framework has been integrated with a recently proposed cross-layer protocol that provides both an acyclic overlay network and knowledge of the neighbors' state. Finally, we compared the performance of our technique with the well-known Practical Network Coding technique combined with probabilistic flooding. We observed that the limitations of the generation size to the number of descriptions, imposed by the delay constraints of real-time video, severely affect the performance of the reference technique, which as a result

is consistently outperformed by the proposed approach. This technique, together with its experimental results, has been presented at the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012).

For the second technique we extended the use of Expanding Window Network Coding to multiple description video in order to guarantee instant decodability to the flow. Our proposed RD-optimized scheduler determines the order in which the frames are included in the coding window. In order to reduce the probability of generating non-innovative packets, the sources operate a classification of the frames (clustering) that provides them with a degree of freedom in the choice of the schedule. We compared the performance of our technique with Expanding Window Network Coding applied on both on Single Description and Multiple Description coding, assuming a trivial scheduling order, and (in the case of MDC) limiting the combinations within the same description. We observed that the introduction of the scheduling, jointly with the possibility of mixing packets across descriptions, significantly improves the performance with respect to the reference techniques, in terms of video quality perceived by the user. This contribution has been presented at the 20th European Signal Processing Conference (EUSIPCO 2012), organized by the European Association for Signal, Speech and Image Processing (EURASIP).

Chapter 3

Network Coding for Multi-view Video Streaming

Contents

In this chapter we discuss another application of network coding in the context of video diffusion. First, in Sec. 3.1, we give an overview of multi-view video and some of the representation and compression schemes found in the literature. We then present in Sec. 3.2 our contribution to the streaming of multi-view video in wireless networks from multiple senders. Finally, in Sec. 3.3 we report the obtained results.

3.1 Multi-view video

In recent years, the advances in video acquisition, compression, transmission and rendering have made it possible for the development of technologies that can enhance the viewer experience by including the third dimension in the visual experience. While the traditional video offers the viewer only a passive way of observing the scene, a more realistic experience can be obtained through applications like 3D video, immersive teleconference and holography. While 3D cinema productions have generated big revenues, other applications

such as 3DTV and Free Viewpoint TV (FTV) [TTFY11] are becoming more desirable due to the increased affordability of 3D displays for home use.

Multi-view video (MVV) is one key point for these applications: it consists in the simultaneous representation of a scene seen by N cameras placed in different spatial positions, called points of view.

A key factor in this context is the way the human visual system perceives depth. While each eye picks up only 2D images, the slight difference in viewpoint is exploited by the brain in order to assess depth and create a 3D model of the scene.

Several techniques have been used to convey depth perception to the viewer. In stereoscopic video, the 3D impression is provided with a pair of left and right videos, each dedicated to the corresponding eye, while the system ensures that the appropriate signals are viewed by the correct eye. Most current 3D television sets use glasses for a polarized 3D system in which the two images are projected superimposed onto the same screen through different polarizing filters. Another 3D system is based on an active shutter in which the glasses are controlled by a timing signal that alternates between darkening the glasses over one eye, and then the other, in synchronization with the refresh rate of the screen. The availability of more views on certain displays allows for a better viewing comfort, as the user can move freely without the need of glasses (autostereoscopic displays).

By using more than two cameras during video acquisition, more texture and depth information is available. This can be used to reduce the occlusion problem (*i.e.*, points that are visible in one image but not the other) found in stereo-view. It can also be used to synthesize virtual views different from the ones acquired. This functionality is used in FTV where the user interactively controls the viewpoint in the scene.

On the other hand, since 3D video could not be deployed if the quality perceived by the user did not exceed the existing 2D quality standards, the bandwidth for storage and transmission of the multiple views is accordingly increased.

3.1.1 Multi-view video representation

The conventional stereoscopic video consists in a pair of videos, left and right. A common way to represent and transmit these two video streams is to multiplex them temporally or spatially. In the time multiplexed format the left and right pictures are interleaved as alternating frames. In the spatial multiplexing the two pictures are squeezed along the horizontal or vertical axis to fit in the original picture dimension, with the loss of spatial resolution along that axis.

In the case of multi-view video, adjacent views act like local stereo pairs to guarantee stereoscopy to the viewer. The multiple views can support head motion parallax within practical limits. Another representation is based on approximating the geometry of the scene in order to generate virtual views.

Another representation, 2D+Z, is based on the use of a *depth map* in addition to the

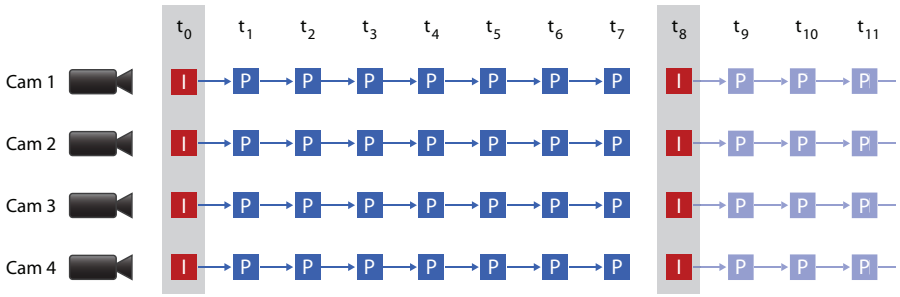


Figure 3.1: Simulcast coding structure.

standard 2D color image (texture). The scene is captured also by range cameras that can compute the distance from the scene to the camera sensors. For example, in structured-light 3D scanners the scene is illuminated with a specific pattern, like a grid of horizontal and vertical lines, from which the geometry of the scene can be computed. A time-of-flight camera emits short light pulses that are reflected by the scene and captured by the camera's optical system. Since the speed of light in air is a known constant, a range value can be obtained in each pixel by measuring the time the light takes from emission to reaching the camera again.

Multi-view video plus depth (MVD) is a combination of the 2D+Z and MVV representations: multiple 2D videos are used with their associated depth video. While this technique can give a more accurate description of the scene and increase the resolution, the drawback is the complexity of the rendering and the volume of the input data.

3.1.2 Multi-view video compression

A first solution for compressing multi-view video, called *simulcast* [MSMW07], is to compress each view independently, reducing the temporal and spatial redundancy by applying video compression (*e.g.*, H.264/MPEG-4 or HEVC [SO10]) on the 2D video sequences independently. While simple to perform, this technique does not take into account the similarities among the views that can be used to further compress the data. On the other hand, it allows for easier switching between views.

The idea of exploiting the statistical dependencies from both temporal and inter-view reference pictures was incorporated in the *Multi-view Video Coding* (MVC) extension of the H.264/MPEG-4 AVC standard [VWS11].

While for intra-view (temporal) prediction the encoders use motion estimation and motion compensation, inter-view prediction for a stereoscopic video is obtained by *disparity* estimation and compensation. The disparity is the displacement of a point projected onto one image with respect to the other image.

With MVC two particular schemes are possible: view progressive or fully hierarchical. In the view progressive architecture, presented in Fig. 3.2, the first view, called the *base*

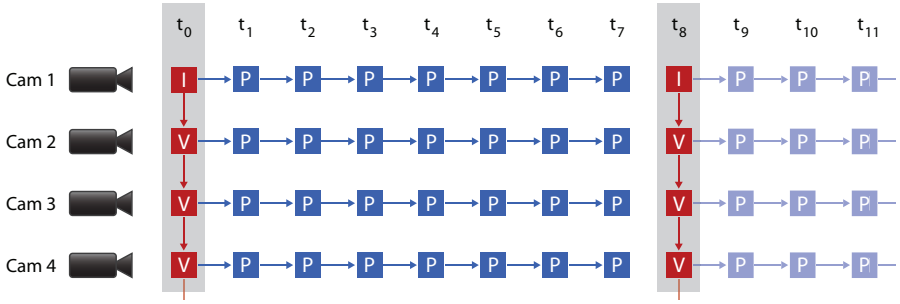


Figure 3.2: View progressive architecture of H.264/MVC.

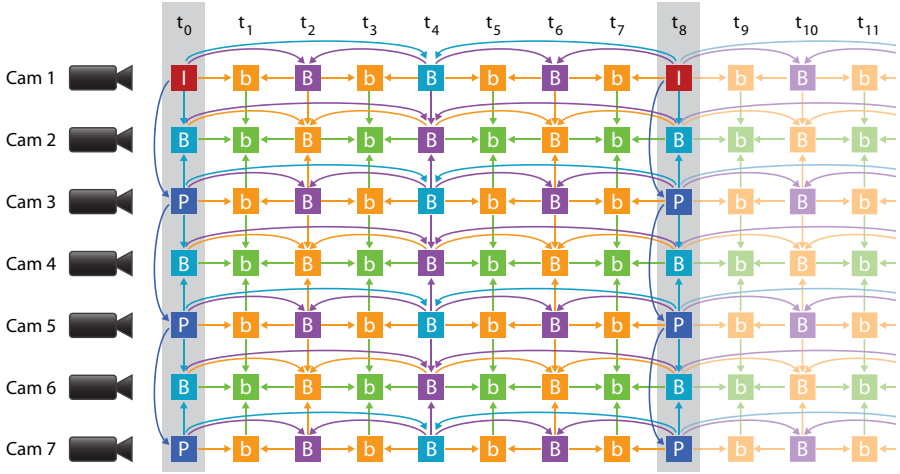


Figure 3.3: Fully Hierarchical architecture of H.264/MVC

view, is encoded independently from the others. In any other view, for each GOP, there is one frame, the V-frame, that is predicted using only inter-view prediction from the corresponding I-frame in the base view. For all other frames only temporal prediction is used.

In the second architecture, presented in Fig. 3.3, both hierarchical temporal prediction and inter-view prediction are performed for all P/B-frames of all views except for the first view (the base view). Therefore, a predictive frame is not only predicted from temporally neighboring frames but also from corresponding frames in the adjacent views, only the base view is encoded independently. This structure does not allow random access to views without completely decoding the entire GOP.

With the view progressive architecture there is a trade off between the simulcast and the fully hierarchical structure. In this structure random access is simpler: only key frames need references from other views to be decoded. P-frames can be decoded without access to other views. The predictions for each frame are performed by using a set of reference frames that can either represent a different time in the same view or the same time in a

different view.

Even though work is in progress within the standardization groups to provide an efficient compression of an even higher number of views with their depth (3D Video Coding, or 3DVC) [MPE11] based on HEVC [SO10], the traffic of multi-view video still remains several times larger than that of traditional video. This adds to the existing problems of current networks, which can be unreliable or have to meet many client demands.

3.2 Proposed contribution

In this section, we detail our contribution, whose objective is to provide robust transmission of multi-view streams over lossy networks, such as wireless networks, with a good trade-off between resiliency to losses and timely delivery.

As seen in chapter 2, network coding with an expanding window (EWNC) [VS10] can be used for the timely delivery of multiple description video over an unreliable network such as a wireless network. Since the wireless medium is inherently broadcast, we exploited the possibility of the receiver being exposed to multiple senders. Our objective was to ensure that the senders choose innovative coding vectors even though they do not coordinate their actions. The results were improved with our *Rate-Distortion Optimized* (RDO) scheduling algorithm that selects which video packet has to be added to the coding window in such a way as to maximize the expected video quality measured at the receiver.

These results have prompted us to tackle the subject of multi-view video. Here, an additional challenge arises, which depends on the prediction structure used. Whereas in MDC, by definition, each description has to be independently decodable, therefore no prediction exists among descriptions, in MVC both in progressive and fully hierarchical architectures inter-view predictions exist that we need to take into account in the scheduling algorithm.

As mentioned in Sec. 2.3, the efficiency of EWNC depends on the order in which the data packets are included in the coding window. The original EWNC method was proposed for a single view layered video, therefore the priority of the packets was naturally imposed by the dependencies among layers.

In our scenario, as we deal with multiple uncoordinated senders sharing a broadcast medium, if they all were to choose the same order of packets (imposed by the layered structure), at any sending opportunity they would send non-innovative combinations.

In general, if a prioritization is optimal, it is also unique, thus all the senders would always transmit dependent combinations, defeating the purpose of using NC. In order to take advantage of the benefits of NC in terms of loss resiliency, we need to generate a variety of schedules, possibly slightly sub-optimal, but with acceptable performances.

The GOP structure of a multi-view video coding technique (such as H.264/MVC) leaves a certain degree of freedom in the scheduling, as frames on the same prediction level can be sent in any order. However, this degree of freedom may not be enough to provide a

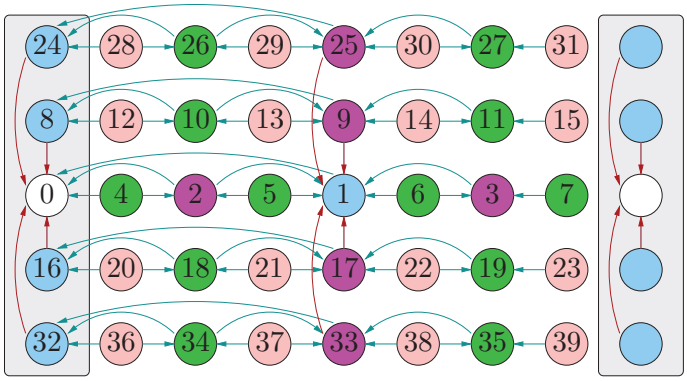


Figure 3.4: Example of MVC prediction structure with intra- (horizontal arrows) and inter-view (vertical arrows) prediction. The numbers indicate the coding order.

sufficient number of different schedules for the different senders.

In a multi-view context, the pool of frames candidate for inclusion in the coding window is again a bi-dimensional *multi-view GOP* (MV-GOP), *i.e.*, a rectangular buffer of size $N \times W$, where N is this time the number of views and W is the temporal size. An example of a simple hierarchical multi-view GOP structure is shown in Fig. 3.4. Notice that the frames in this buffer are not ordered by their play-out date, but in encoding order, so that frame dependencies are respected.

The task of the scheduler is to provide an order in which the frames in the MV-GOP are included in the coding window. Since wireless networks are affected by churn and mobility, and the video stream can be interrupted at any moment, it is desirable that any new combination maximizes the marginal benefit in terms of RD properties. In other words, at each step, we want the scheduling algorithm to select the frame that optimizes an RD criterion for insertion in the coding window.

However, corresponding frames of different views have differences in their RD properties, which would lead to a unique optimal policy of inclusion in the coding window.

In order to solve this problem, similarly to the technique proposed in Chapter 2, we cluster the video frames. The frames are classified based on their RD properties, a task that takes place at the video source, after the video encoding and before scheduling for transmission. Frames with similar RD points are assigned to the same cluster; each frame is labeled with the average rate and distortion of its cluster, possibly quantized.

The labels are decided *only once* at the encoder, where rate and distortion are known with negligible computational overhead.

The purpose of clustering is to increase diversity between senders policies. At the intermediate nodes, at each sending opportunity, the scheduler selects for inclusion in the coding window a frame f^* among the eligible ones, *i.e.*, those whose references for prediction, if any, are already in the coding window. The selected frame minimizes a

rate-distortion cost function:

$$f^* = \arg\min_{f \in \mathcal{F}} \{J(f) = R(f) + \lambda D(f)\}.$$

However, nodes use the rate and distortion values reported on the labels to evaluate the cost function, rather than the actual values. It is therefore very likely that several frames appear to have the same cost. In this case, a node would randomly select one of them. This frame is added to the coding window, increasing its size by one. The size of the coding window is reset to zero with the new MV-GOP, similar to Algorithm 1 in Chapter 2.

Notice that large clusters increase the chance of different nodes selecting different frames, thus reducing non-innovative packets. On the other hand, if clusters are chosen too large, the scheduler will randomly choose among frames with very different values of the cost function, resulting in a sub-optimal performance.

Ideally, the size of the clusters should be chosen according to the expected number of senders that are going to transmit at the same time (so that each sender could select a different frame of the same cluster), which can be roughly estimated with the node density of the network.

In practice, clustering can be performed in several ways. For instance, a coarse but simple scheme is to assign all the frames on the same prediction level to a single cluster. This scheme is independent from the actual RD properties of the sequence and can be easily implemented; nevertheless, it can be quite efficient if the views have frame-by-frame similar RD properties. If the corresponding frames of different views have unbalanced properties, then a more sophisticated scheme can be employed.

An example of two different scheduling orders is presented in Fig. 3.5. For the sake of clarity, only the scheduling for the first 20 packets is presented. We can observe that, if only a subset of a cluster is chosen, the two schedulers choose different frames within it. If the whole cluster is chosen, then the frames still differ in the order they are included in the coding window.

As mentioned before, the prediction structures in the case of MVC give raise to a challenge in the clustering that did not occur in the case of MDC, given the extra dependencies among views. We use a simple hierarchical structure (Fig. 3.4) which provides a better compression efficiency than a progressive architecture but less entanglement than a fully hierarchical structure, thus allows for higher diversity in the formation of clusters. In our technique, the throughput is a function of the innovativity of the linear combinations, hence having more frames with the same prediction level allows for larger clusters. The multiple senders have more options in their choice of frames within the cluster.

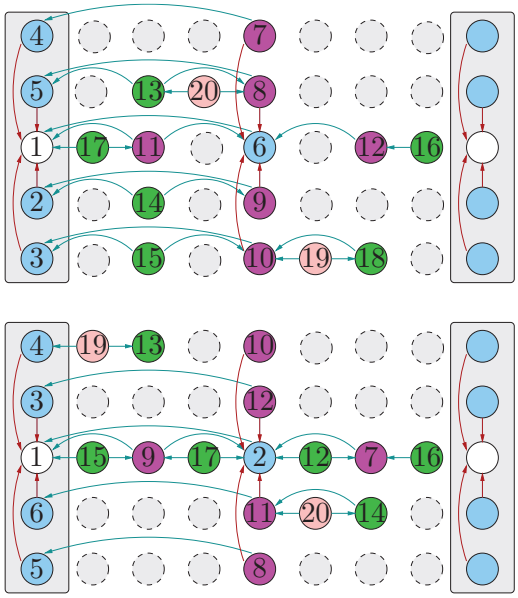


Figure 3.5: Two possible schedules (first 20 rounds). The numbers indicate the round in which the frame is included in the coding window. The dashed border identifies which frames have not been selected yet for inclusion in the coding window at the 20-th round.

3.3 Experimental results

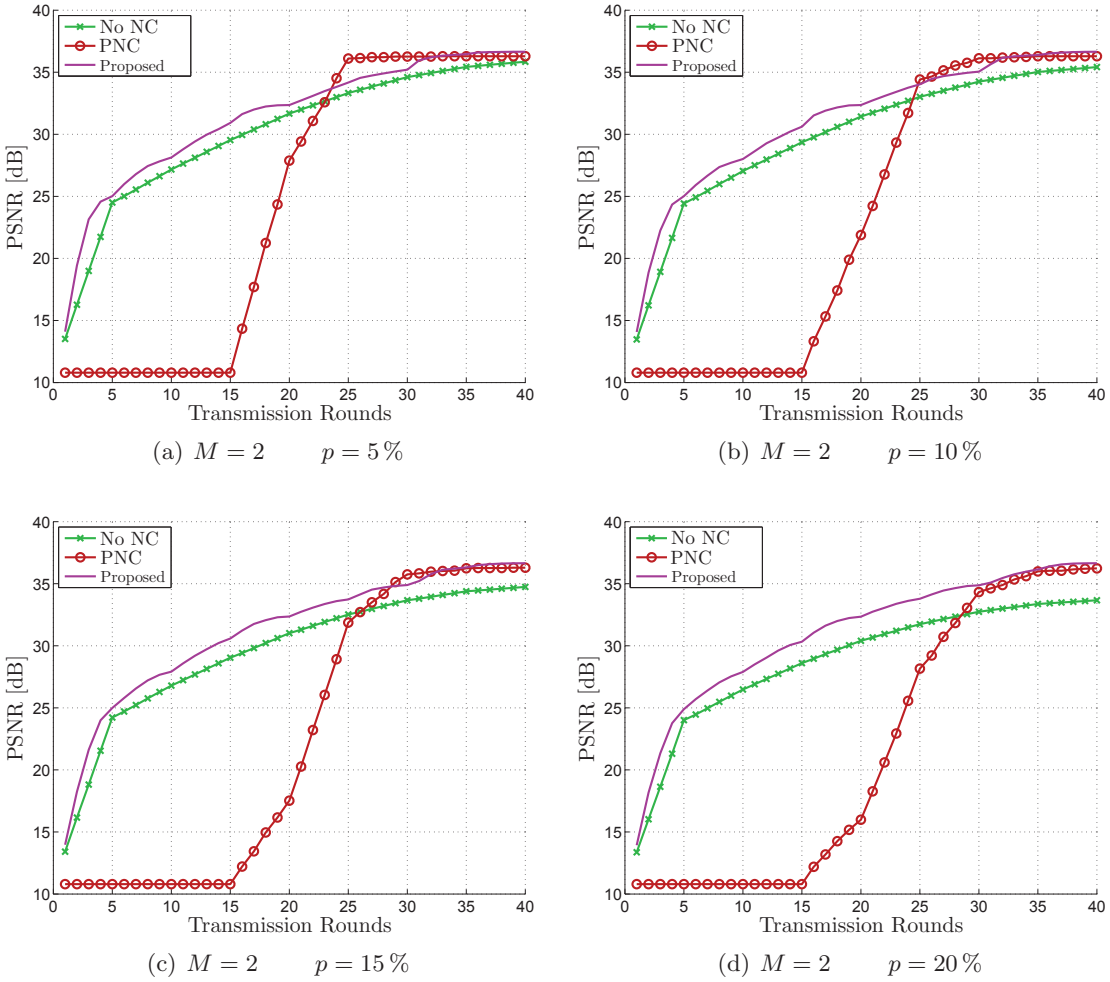
In the following, we present the results of the proposed technique and compare them with the results achievable using PNC on each view independently. For reference, the results achieved without using NC are also presented.

In our scenario, M sources transmit a multi-view video sequence to a single receiver R. In the simulations, the video sequences "ballet", "bookarrival", "breakdancers", and "doorflowers" (5 views, 1024×768, 100 frames) have been encoded in H.264/MVC using the GOP structure in Fig. 3.4 with QP 31, 34, 37, and 40.

In order to allow a clear evaluation of our technique, a discrete-time transmission model is assumed: the time is segmented in *transmission rounds* wherein each source sends exactly one packet from a predetermined transmission buffer. Each channel between a transmission buffer and the receiver buffer is in general lossy, with independent uniform packet loss probability p_m; the transmissions on different channels do not interfere with each other. At the end of each round, the receiver decodes all the frames available in its buffer, generating a reconstructed sequence. Henceforth, for all MVC techniques the PSNR of a sequence will be computed as the average over the views, unless stated otherwise.

In Figure. 3.6, we report a comparison with the reference techniques for a two senders scenario with packet loss probabilities of 5, 10, 15, and 20 %.

We observe that, if no network coding is used, each received packet increases the PSNR. However, the transmission cannot recover from losses, thus in scenarios with high loss probability, the maximum quality is not achieved. For instance, in Figure 3.6(d) we



(a) $M = 2$ $p = 5\,\%$ (b) $M = 2$ $p = 10\,\%$

(c) $M = 2$ $p = 15\,\%$ (d) $M = 2$ $p = 20\,\%$

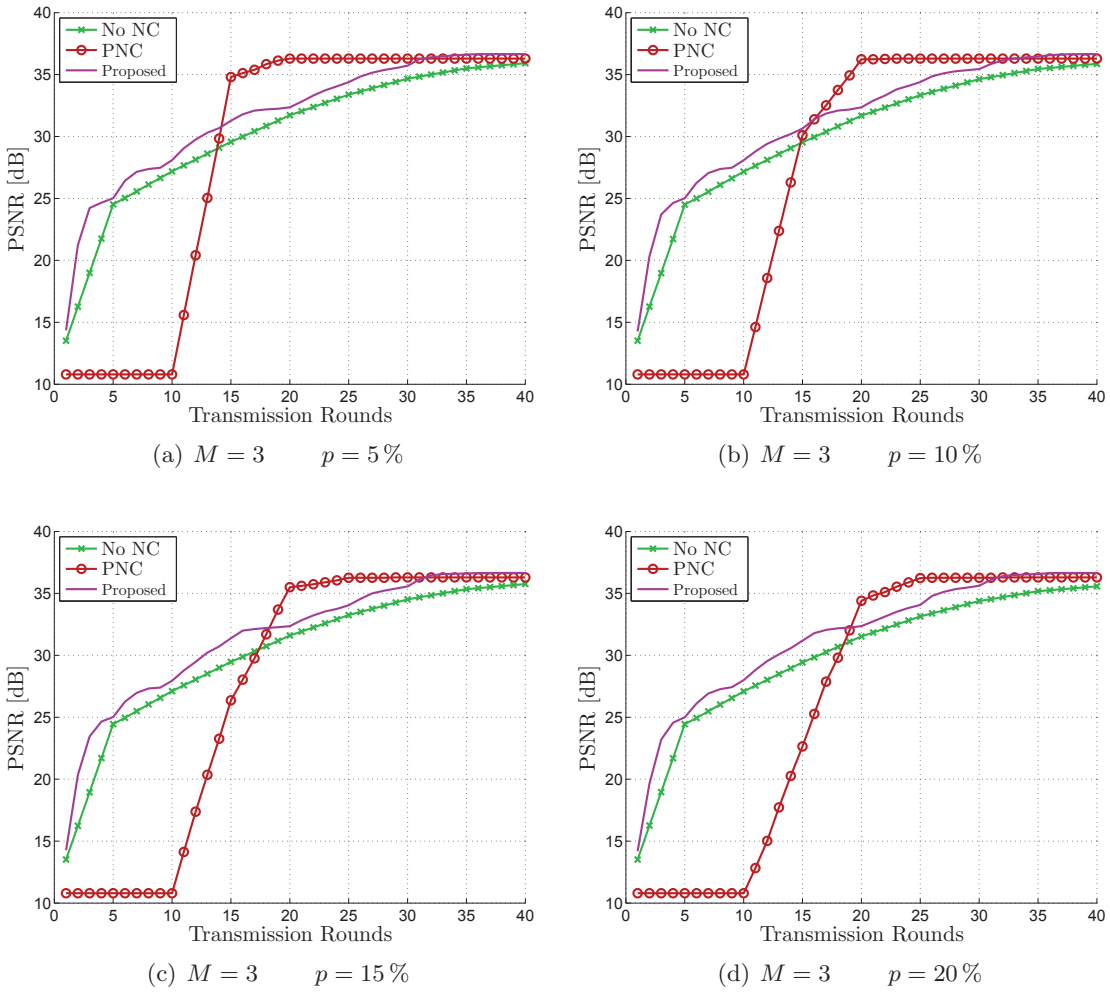
Figure 3.6: Comparison of the average PSNR of the decoded sequences, for $M = 2$ sources and packet loss probability $p_m = p$ for all channels. For each sequence the PSNR is computed as the average over the views.

see that in the two-senders scenario a 20 % packet loss probability reflects in a PSNR loss of more than 2 dB.

Conversely, PNC eventually achieves the maximum quality, but the receiver cannot decode any frames in the first rounds. For instance, as we can see in the figure, the PNC technique is unable to decode any frame before the 15-th round, and does not achieve a better video quality than the proposed technique in less than 23 to 34 rounds, depending on the packet loss probability.

The necessity of buffering for a long time is an undesirable property in a wireless environment, as the communication could be interrupted at any moment, leaving the node with no useful data. Also, when uncoordinated, all senders will first transmit packets belonging to the base view, which will reduce the diversity of the packets received by the decoder, thus increasing the delay if no method for early decodability is used.

In Figure 3.7 we present the results for the same packet loss probabilities in a three



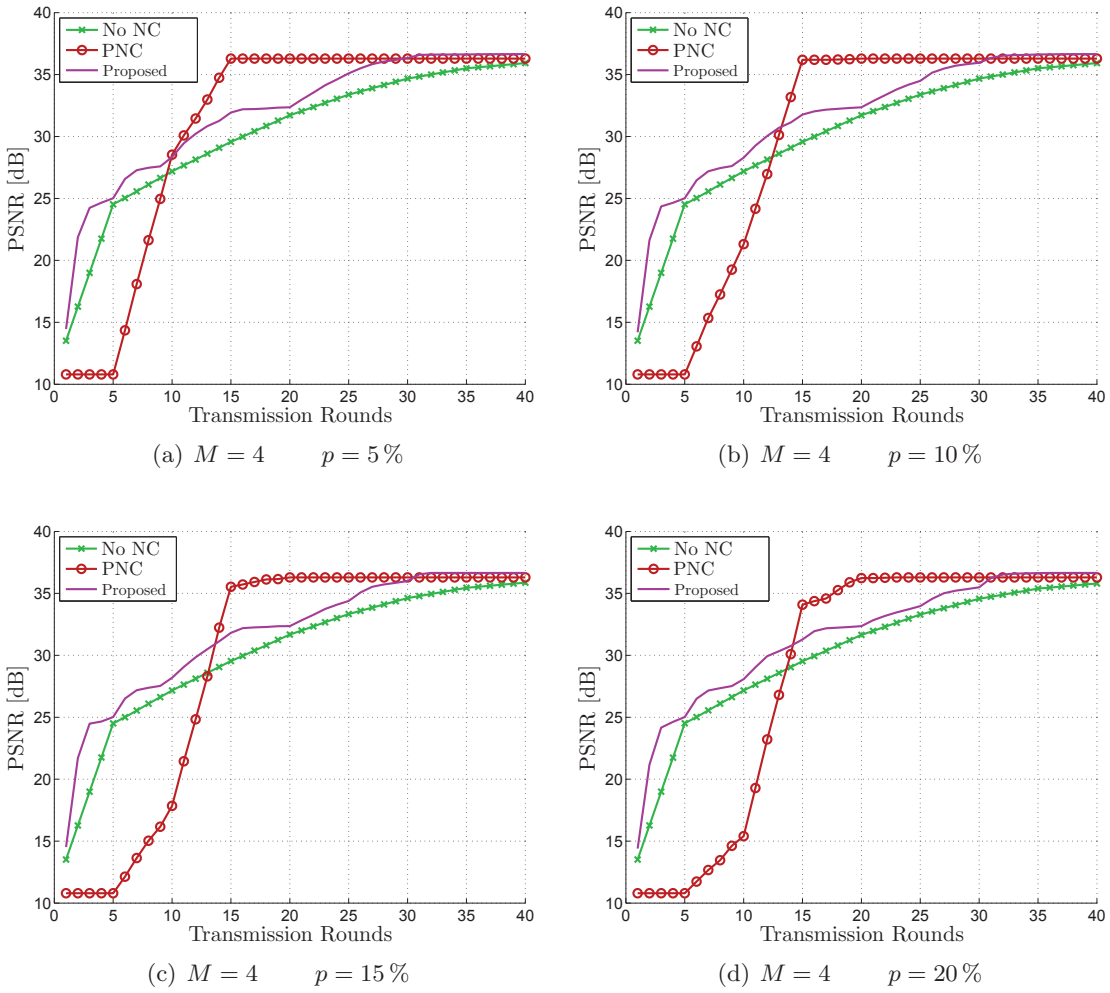
(a) $M = 3$ $p = 5\,\%$ (b) $M = 3$ $p = 10\,\%$

(c) $M = 3$ $p = 15\,\%$ (d) $M = 3$ $p = 20\,\%$

Figure 3.7: Comparison of the average PSNR of the decoded sequences, for $M = 3$ sources and packet loss probability $p_m = p$ for all channels.

senders scenario. As we can see, while the performances of PNC improve slightly in the sense that the receiver starts decoding the stream a few transmission rounds earlier. However, the results still show a consistent and clear advantage of our technique in the first 10 to 20 rounds (depending on the packet loss probability). Furthermore, the non-coding technique also benefits by a higher number of sources since the duplicated packets can partially compensate for the packet losses. However, when the packet loss rate is higher than 10 % this effect is not sufficient to provide eventually the full video quality.

These effects can also be seen more clearly in Figure 3.8, where a four senders scenario for the same packet loss probabilities is presented. In this case, the PNC technique starts decoding quite early and can surpass our proposed technique in 10 to 15 rounds, which could be acceptable in some applications, but is still a non-negligible delay. Similarly, the non-coding technique is able to exploit the higher number of senders by compensating for more losses even though in the first rounds it can still perform about 5 dB less than our



(a) $M = 4$ $p = 5\,\%$ (b) $M = 4$ $p = 10\,\%$

(c) $M = 4$ $p = 15\,\%$ (d) $M = 4$ $p = 20\,\%$

Figure 3.8: Comparison of the average PSNR of the decoded sequences, for $M = 4$ sources and packet loss probability $p_m = p$ for all channels.

proposed technique.

To summarize, we can conclude that our approach, thanks to the early decodability offered by EWNC and the variety in the scheduling provided by the clustering, is able both to provide a consistently better video quality than a non-coding technique and to achieve an acceptable video quality of the decoded sequence in a fewer number of transmission rounds than PNC. These benefits of the proposed technique are more visible when the transmission conditions are harsher, *i.e.*, when fewer senders provide the content and packet loss rate is higher.

3.4 Conclusions

In this chapter, we presented our proposed technique for streaming multi-view video content over unreliable channels using network coding. The key idea is to use Expanding

Window Network Coding in order to guarantee instant decodability of the flow. The frames are included in the coding window in an order determined by an RD-optimized scheduler. In order to reduce the probability of generating non-innovative packets, the sources operate with a simplified probabilistic RD model that provides them with a degree of freedom in the choice of the schedule.

We compared the performance of our technique with Practical Network Coding applied on each view independently, and transmission without network coding, both assuming a trivial scheduling order.

We observe that the introduction of the scheduling, jointly with the possibility of mixing packets across views, significantly improves the performance with respect to the reference techniques, in terms of video quality perceived by the user.

The results we obtained suggest that further research in this direction could be promising, in particular in the direction of a joint design of an overlay management protocol that could select which nodes of the network should rely the stream. The results were reported at the 2012 SPIE Visual Communications and Image Processing (VCIP) conference.

Chapter 4

Distributed Social Caching using Network Coding

Contents

In this chapter, we address the topic of cooperative caching of multi-view video content, wherein users who recently acquired the content contribute parts of it by providing a distributed cache service for the benefit of their social group. First, in Section 4.2, we give an overview on the related work. Then, in Section 4.3 , after having presented the system model, detailing and motivating our assumptions, we describe the selection method used to decide which frames of the content will be included in the combinations stored in the local cache of the nodes. In Section 4.4, we present the experimental validation of the proposed technique, along with a comparison with a network coding technique that does not take into account the users' preferences, and analyze the results. Finally, in Section 4.5 we draw our conclusions and point out to some future work.

4.1 Introduction

Cooperative networking is a topic that has been intensively investigated in the literature as an alternative to the traditional client-server paradigm, as it allows an alleviation of the load on the storage equipment and a better use of the network capabilities. Cooperation takes place in many content distribution scenarios, such as peer-to-peer networking

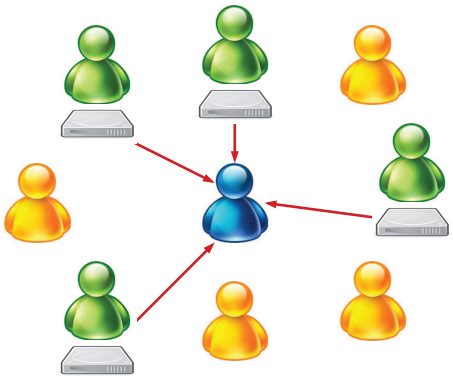


Figure 4.1: Example of social caching. Rather than accessing an external server, a user will try to collect as much data as possible from other members of its social group that have already viewed and cached the content.

(P2P) [AML+05] and mobile ad-hoc networking (MANET) [FJL00], where users have to contribute some of their resources in order to improve the overall performance of the system [PS02].

While in the general case users tend to be selfish, and the system has to set incentive mechanisms to keep the users engaged, this is not the case when cooperation takes place within a social network. Users are more willing to share their resources for the benefit of their friends or to gain recognition in their social network [FHPK13].

One context that could greatly benefit from the advantages of cooperative networking is multimedia streaming, and in particular, the 3D video applications and Free-Viewpoint TV (FTV) [TTFY11] mentioned in Chapter 3, as they require much more bandwidth than conventional content both to store and to transmit the multiple views of the scene.

As we mentioned in the previous chapter, several techniques for *Multi-view Video Coding* (MVC) [VWS11] have been proposed to exploit the statistical correlation of the source, both temporal and spatial, to achieve a better compression and to allow for interactive switching among the views [COC11, PCDPP11, SBB+12].

In this chapter, we study a novel scenario in the context of live event interactive multi-view. Namely, in our scenario, the users form a *social group*, within which they share and exchange multimedia content in the form of multi-view sequences. In particular, due to the nature of the social interactions, a content acquired by a user will be most likely requested by other members of the group within a relatively short period of time.

For the benefit of the group, each user who has already acquired the content will be encouraged to keep a part of it in a local cache, so that when a fellow group member will request the content, it will be served by accessing the group cache rather than an external server. This will also be beneficial to the server itself, since part of the load will be handled by the social group in a cooperative fashion.

Thus the group's content cache is distributed, *i.e.*, the amount of data available to a member is the sum of the amounts available on every group member. However, each

member would like to reserve only a limited amount of memory on its terminal to dedicate to the group. It is therefore crucial to make the best use of this space, by minimizing the amount of memory that is wasted, *e.g.*, on duplicates. Also, it is highly desirable that this optimization takes place with as little communication among the users as possible —ideally none, in order to reduce the additional network traffic [KLL+97, BCF+99].

Here, we present our scheme in which the users of the social group, in order to provide a way for the members of the group to obtain a multi-view content without the need to access a remote server, store a selection of the frames they access, selected to minimize the distortion of the stream. This minimization will be performed by taking into account the preferences of the group members in terms of displayed views, which have already been successfully exploited for video caching in the context of single-view streams in mobile environment [PRB+09].

4.2 Related work

The particular coding structures of the multi-view representation reflects in a non-trivial impact of each coded frame in the overall quality of the reconstruction of the multi-view content. If this impact is properly captured, it can be used to design an intelligent transmission scheme that allocates the limited network resources in a rate-distortion optimized order (scheduling). In order to effectively disseminate the content to the end-users, an analogous scheme can be devised for scheduling the frames for inclusion in a distributed cache [GR05].

For instance, in the works of Huang *et al.*, the access cost –*i.e.*, the cost that a user incurs in when it has to contact the content server directly, since the content it is requesting is not available in the distributed cache– is minimized by replicating parts of a multi-view stream through a network of content distribution servers with heterogeneous storage capacities [HCCF12, HZC+12]. Of particular interest is the fact that the technique proposed by Huang *et al.* produces replication schemes able to support view-switching during an interactive multi-view session. The authors propose for this problem both a purely heuristic solution for the single movie case [HCCF12], and a heuristical relaxation of the problem, that provides near-optimal solutions with bounded error, and that accounts for the multiple movies case [HZC+12].

Network Coding, which has been discussed in detail in Chapter 1, has also been proposed as an elegant and effective solution to distributed storage and transmission: rather than caching the data packets, the users will generate and store random linear combinations of the packets, discarding the original content. The advantage of this technique is that even though the users act independently from each other, with high probability each of them will contribute innovative information to the distributed cache [GR05, HRM11].

This form of replication scheme –that stores and transmits linear combinations of the video packets rather than the packet themselves– presents several advantages over

traditional non-coding techniques. First, it enhances cache diversity, in the sense that different storage nodes will most likely store innovative data, and furthermore, it simplifies the location of the content because it eliminates the need to locate specific parts, due to the fact that any number of innovative parts will be sufficient to retrieve the whole content.

One technique based on the nework coding replication principles has been proposed by Wang *et al.* [WZLJ10] for peer-to-peer video-on-demand applications. More recently, Kao *et al.* [KLWK12] also proposed a more general framework able to provide an interactive streaming service, *i.e.*, allowing random access operations to the users. However, neither of these techniques address the multi-view case nor take into account the rate-distortion properties of the stream.

Other existing works have tackled the subject of distributed video caching, achieving similar properties, by proposing to use rateless codes –conceptually similar to network coding– for video delivery [GDMC11, GSD+12]. However, unlike our scenario, these studies address the problem in the context of wireless networking, which have very different characteristics and constraints, as we have seen in Chapter 2.

Furthermore, even though these techniques have been proposed for video delivery, only the delay requirements of video streaming have been exploited, while our method is tailored for multi-view video content and in particular it uses the prediction structure of the encoded sequence in its optimization algorithm. It should be noted that in our method, rather than storing a selection of complete content streams, we store a selected part of a multi-view stream. In other words, rather than a simple hit/miss rate optimization, a proper rate-distortion optimization is performed in order to provide the users with the best possible video quality given the limited storage space allocated to the caching service.

Unlike previous works on multi-view streaming, rather than focusing on the encoding of the content, and rather than considering each client as an independent agent, we study how the distribution of the stream can take advantage of the relationships that exist among the different clients.

In the context of multi-view, techniques based on the users' preferences have already been applied to rate allocation in source coding [FCF10]. However, we consider here a completely different problem, in which the multi-view video has been already encoded, and we must decide about the parts of the content that are to be stored on the distributed cache.

We propose to apply similar principles in the selection of the parts of the content that will be stored on the distributed cache.

4.3 Proposed contribution

In this section we describe our proposed method of cache selection and encoding for a distributed cache of multi-view video content based on the users' preferences.

4.3.1 System model

We address the topic of distributed caching for multi-view video content. In order to optimize the rate-distortion performances of the stored content, we select the frames to be stored based on their popularity among the users.

Before explaining in detail our proposed technique, here we list and justify some assumptions about the system that will be used in the design of the technique.

- From the point of view of the network, we assume that the users are connected in a mesh network. This reflects our definition of social group wherein each member is connected to every other. Furthermore, we assume that the connectivity among the users is not an issue, *i.e.*, that the channel connecting the users provides sufficient capacity for transferring the whole multi-view stream, whereas the bottleneck is situated between the users and the server. This justifies the use of the caching system, and reflects a situation in which the group members are either physically closer to each other than to the server, or are more willing to provide more uplink resources to their fellows than the server is.

- From the point of view of the content, we assume that the stream is encoded using H.264/MVC [VWS11] or a similar inter-view prediction scheme. In particular, the stream is encoded using the prediction structure described in Chapter 3 and depicted in Figure 3.4. This structure uses inter-view prediction in order to achieve a better coding efficiency, but is not fully hierarchical in order to reduce the dependencies among the frames, thus reducing the propagation of the effects of losses.

- For the user's preferences, we assume that the choice of the preferred view for each user follows the same, known distribution. This is consistent with the dynamics of a social group: members of a social group customary share opinions and preferences with each other, and a well-designed system reflects this behavior. Notice that how the learning and keeping track of the popularity distribution take place is outside the scope of this article, and shall not be addressed in the following.

- Finally, we assume that the preferences distribution changes slowly over time. Here slowly means that the distribution can be considered valid for at least the duration of a GOP. This implies that our system is able to work even when users' preferences change as frequently as once per GOP, which typically lasts a fraction of a second. Any change in preferences during a GOP will be taken into account at the next GOP.

4.3.2 Proposed method

As mentioned in Sec. 4.2, user preferences have already been used to optimize the rate allocation in the encoding process. Here, we show how they can be used to decide which

N	Number of views
M	Number of frames per view
$\mathbf{B}$	Bi-dimensional frame buffer
$\mathbf{b}$	Caching decision
$\mathbf{p}$	Users' preferences distribution
$\mathcal{W}$	Coding window
R	Actual size of the coding window
R_v	Number of frames of view v in $\mathcal{W}$
D	Expected total distortion
D_v	Distortion of view v

Table 4.1: Summary of the used notation.

parts of the content have to be stored in the distributed cache in order to optimize the rate-distortion properties of the stored stream.

In our technique, the cache operates using an implementation of Practical Network Coding (see Chapter 1). The content is divided into groups of packets and only packets belonging to the same group can be mixed together. In our system each packet contains only one encoded frame, and we only mix frames belonging to the same GOP.

As seen in the first chapter, by using PNC, the users can store linearly independent combinations without need of coordination. Thus, using PNC simplifies the task of retrieving the content, as a generic request can be broadcast to all the members of the group without the need to locate specific parts of the content and each member will respond with a combination from its cache.

For content that requires high bit-rates, such as multi-view video content, the total memory available on the distributed cache may still be insufficient to store the whole content, especially for small groups with limited resources. Thus, a challenge arises in deciding which parts of the content it is more sensible to store. Our intuition is that, in the case of multi-view, a great source of information lies in the *preferences* that each user has towards a specific view. These preferences depend of course on the content, but due to the nature of the social relationships among members, who share common interests and preferences, we can assume that they are quite uniform within the group.

We model the distribution of users' preferences with a probability vector $\mathbf{p}$, such that p_v is the probability that a member of the group chooses to watch view v.

The cache will contain packets generated by linearly combining frames belonging to the same GOP. In order to select which frames will be included in the coding window, which we denote by $\mathcal{W}$, we proceed as follows. For each GOP, all the frames of the current GOP are stored in a bi-dimensional frame buffer $\mathbf{B}$, with N rows, and M columns, where N is the number of views and M is the per-view time-length of the GOP. The maximum possible size of the coding window will be NM, *i.e.*, the size of the GOP, but the actual size of the coding window, which we denote R, will typically be smaller.

The organization of the bi-dimensional buffer corresponding to the prediction structure

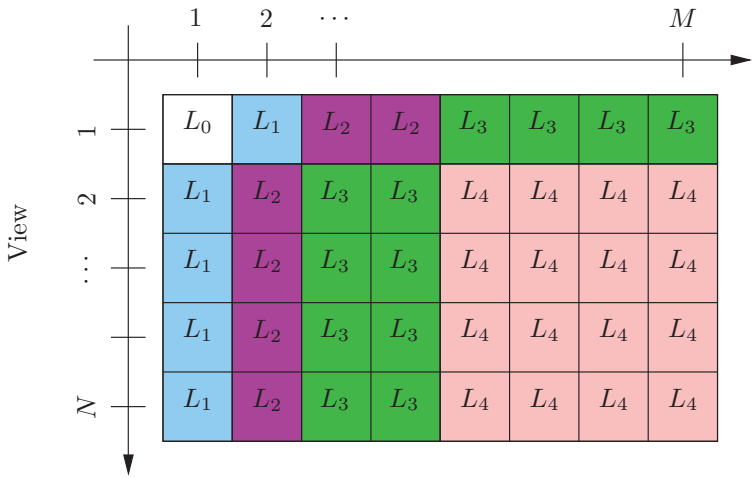


Figure 4.2: Buffer $\mathbf{B}$ for $N = 5$ views and $M = 8$ frames for the prediction structure in Fig. 3.4 of Chapter 3. In each view, frames are ordered by prediction level, then by descending impact on the total distortion of the view.

described in Sec. 4.3.1 and depicted in Fig. 3.4 of Chapter 3 is shown in Fig. 4.2. Notice that the views are re-arranged to reflect the coding order, so the central view corresponds to view 1 in Fig. 4.2, as the other views are predicted upon it.

There is a trade-off to consider in the choice of R: on the one hand, a larger coding window implies a larger share of the content available in the cache, which will in turn increase the video quality perceived by the users. On the other hand, it is inherent in the use of PNC that all the frames in the coding window are decoded jointly. This means that each user has to receive at least R linearly independent combinations in order to be able to decode *any* frame at all. With a large value of R, if too few group members are on-line at a given time, and each one shares only a few combinations, those who access the cache may not receive enough combinations.

Let us consider a vector $\mathbf{b} \in \mathcal{B} = \{0,1\}^{NM}$ that represents a caching decision, *i.e.*, $b_{(v-1)N+k} = 1$ if and only if the k-th frame of the view v is selected to be included in the coding window to be combined with the others and stored on the cache. Our objective is to minimize the expected distortion given a number $R \leq NM$ of frames to be included in the coding window.

Let $D_v(\mathbf{b})$ be the distortion of the view v when the frames selected in $\mathbf{b}$ are available. Note that, due to the inter-view prediction, the functions $D_v(\cdot)$ depend on all the selected frames of all views. For a generic user, the expected total distortion D is:

$$D(\mathbf{b}) = \sum_{v=1}^{N} p_v D_v(\mathbf{b}) = \mathbf{p}^\top \mathbf{D}(\mathbf{b}), \tag{4.1}$$

where vector $\mathbf{D}(\mathbf{b})$ is such that its v-th component is $D_v(\mathbf{b})$.

Algorithm 2 Caching decision algorithm. For each GOP, selects which frames have to be included in the coding window.

```
1: procedure SelectFrames
2:     for all GOPs do
3:         W ← {B_{c,1}};                          ▷ Key frame of the central view.
4:         for i ← 2 to B do
5:             v ← NextView(p);
6:             V ← {f | f ∈ v ∧ f ∉ W};
7:             if V ≠ ∅ then
8:                 F ← first(V);
9:                 while Ref(F) ⊈ W do
10:                    R ← {f | f ∈ Ref(F) ∧ f ∉ W};
11:                    F ← first(R);
12:                end while
13:                W ← W ∪ {F};
14:            end if
15:        end for
16:    end for
17: end procedure
```

The optimization problem can be therefore stated as:

$$\mathbf{b}^* = \arg\min_{\mathbf{b}\in\mathcal{B}} \left\{ \mathbf{p}^\top \mathbf{D}(\mathbf{b}) \right\} \qquad \text{s.t.:} \sum_{k=1}^{NM} b_k \leq R \tag{4.2}$$

In order to solve the problem, first we organize the frames in the buffer in Fig. 4.2 in such a way that, from left to right, the prediction level of the frames is non decreasing, *i.e.*, a frame cannot depend on another frame on its right. Also, within the same prediction level, frames are ordered based on their impact on the total distortion of the view: frames that, when missing, lead to a higher distortion are placed left of those with a smaller impact. This ordering automatically assures that the intra-view dependencies are satisfied if frames are considered in a left-to-right order.

Also, within the same prediction level, frames are ordered based on their impact on the total distortion of the view: frames that, when missing, lead to a higher distortion are placed left of those with a smaller impact. This ordering automatically assures that the intra-view dependencies are satisfied if frames are considered in a left-to-right order.

The problem is now reduced to determining, for each view v, how many frames $R_v = \sum_{i=1}^{M} b_{(v-1)N+k}$, with $R_v \leq M$, have to be included in the coding window, taking into account only inter-view dependencies, *i.e.*, ensuring that for each frame in the coding window, all the frames in other views it is predicted upon are also available.

It is intuitive that the optimal choice of the values R_v is such that $R_v \approx p_v R$. In other words, we should select, for each view, a number of frames that is proportional to its popularity.

Therefore, in order to select the frames, we generate a pseudo-random (and therefore, equal on every node) succession of view indices, distributed following $\mathbf{p}$. The pseudo-random sequence is a deterministic sequence such that the relative frequency of the view indices matches the distribution of $\mathbf{p}$.

The caching decision algorithm works on each GOP independently. For each GOP, the algorithm performs $(R-1)$ iterations, selecting one frame per iteration (the key frame of the central view is always included) for inclusion in the coding window $\mathcal{W}$.

At each iteration i, we consider an index v from the sequence. We check the first (left-most) frame F of view v not yet included: if its inter-view references, denoted as a set $\textsc{Ref}(F)$, are already in the cache, we include it, otherwise we include the first of its inter-view references, where first means the one with the lowest prediction level. If all the frames of view v have been included, we skip to the next index. The algorithm is iterated until R frames are selected. This process achieves very good rate-distortion performances, since both the content characteristics and the user preferences are taken into account when selecting the frames to store in the cache.

Once the coding window has been selected, each node generates a set of packets containing linearly independent random combinations of the R frames. A number of packets between 1 and R will be stored on each node, while the original stream will be discarded. How many packets (combinations) per GOP will be stored depends on each node's resources and its willingness to contribute them to the group.

When a new node wants to access the content, it will broadcast a request to its groupmates. These will answer with as many combinations as they are caching. If enough independent packets are received, the user will be able to decode the whole cache. It will then in turn generate new combinations to contribute to future requests.

4.4 Experimental results

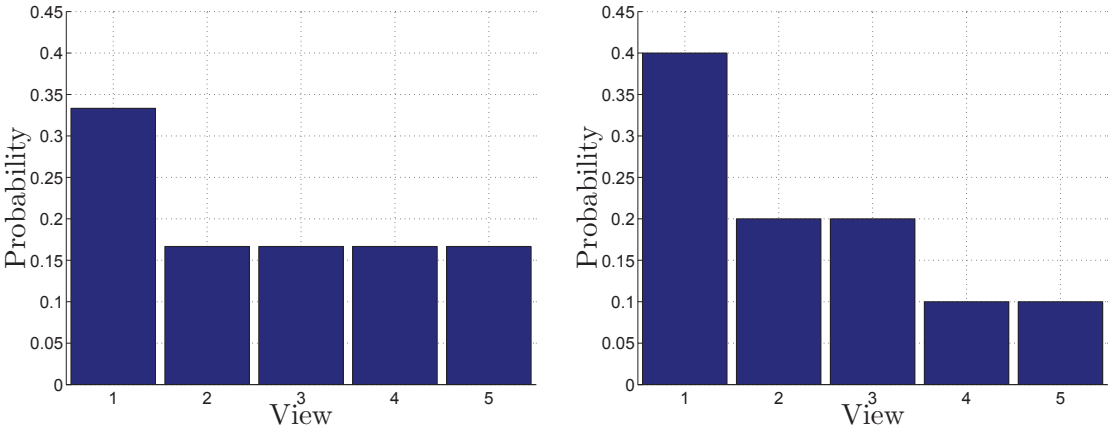
In the following, we present the results of the proposed technique and compare them with three different reference techniques. In our scenario, 100 nodes obtain the multi-view content from the distributed cache. Each user, for each GOP, randomly selects a view according to the distribution of $\mathbf{p}$.

The three reference techniques are chosen to evaluate the contributions of our technique in isolation. In all techniques, the frames are chosen for storage in the cache based on their impact on the total distortion.

- Our first reference uses network coding to store and transmit the cached content, but does not use the users' preferences in its minimization of the expected distortion, *i.e.*, the expected distortion is computed simply as the average distortion over all the views. This technique is inspired by purely network coding techniques used in video-on-demand replication, such as the one proposed in [KLWK12].

"Ballet"	"Bookarrival"
"Breakdancers"	"Doorflowers"

Table 4.2: Multi-view sequences used in the simulations (5 views, 1024×768, 100 frames).



(a) Model M1 (b) Model M2

Figure 4.3: View preference distribution models.

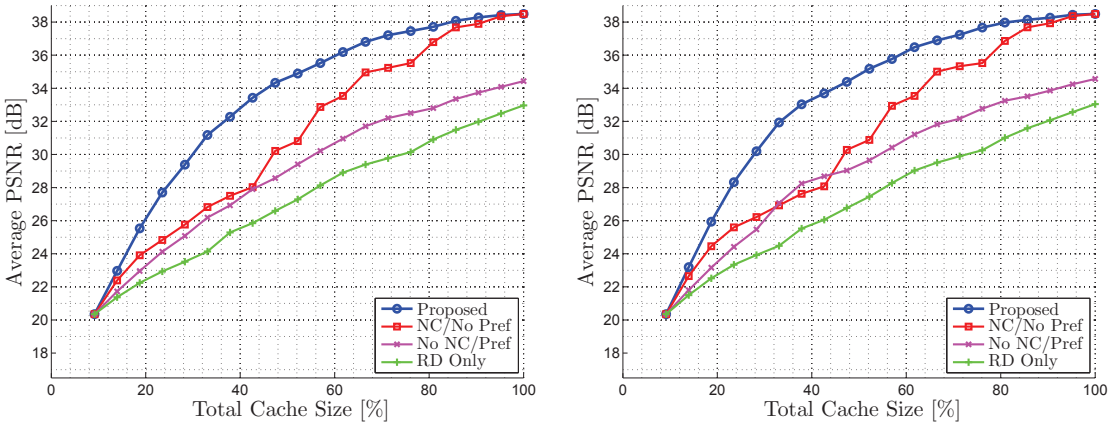
- The second reference uses the users' preferences to optimize the selection of the parts of the video content to be replicated on the distributed cache, but does not use network coding in the storage or the transmission. This technique is inspired by purely preference-aware techniques used in video replication, such as the one proposed in [PRB+09].

- The last reference does not use network coding, nor is aware of the users preferences. This technique is inspired by classical replication schemes, such as the one proposed in [HCCF12].

Notice that in our simulation we measure the video quality as the PSNR achievable by the users decoding the parts of the content they have been able to retrieve by the distributed cache only. Usually, in a practical setup, a user unsatisfied with the video quality could access the central content server to retrieve complementary parts of the stream to increase the video quality, thus incurring a certain access cost. However, we are interested here only in the video quality achievable by making no access to the central server, and access cost is outside of our scope.

In all the simulations, the four video sequences reported in Table 4.2 (5 views, 1024×768, 100 frames) have been encoded in H.264/MVC using the GOP structure described in Chapter 3 and depicted in Figure 3.4.

We tested the system with two different models of view preferences: M1 and M2, depicted in Figure 4.3. In both models, the central view has a higher probability p_c of being selected by the users. In M1 the remaining views are uniformly distributed, while



(a) Preference Model M1.

(b) Preference Model M2.

Figure 4.4: Comparison of the proposed technique with the references for QP= 31 and $p_c = 0.66$. The average PSNR of all the sequences is plotted against the total cache size expressed as a percentage of the content size.

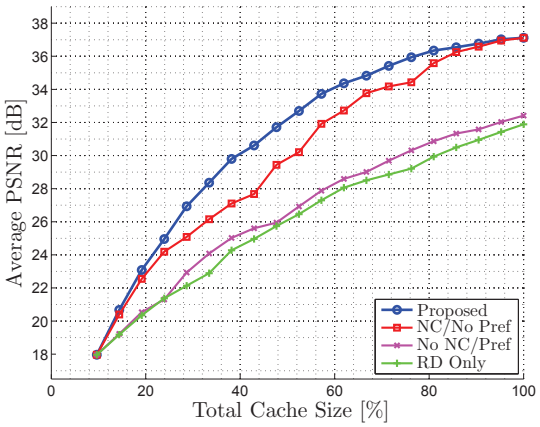
in M2 the views closest to the central are preferred twice as much as the farthest ones. For each GOP, each user randomly selects a view according to the distribution of $\mathbf{p}$ and measures its PSNR.

In Figure 4.4 we show the results of the comparison of our proposed technique with three reference techniques for a QP= 31 and $p_c = 0.66$, for both preference models M1 and M2. On the x-axis we report the total size of the distributed cache expressed as a percentage of the content size, while on the y-axis we report the average video quality –in terms of PSNR– over all the sequences as received by the users.

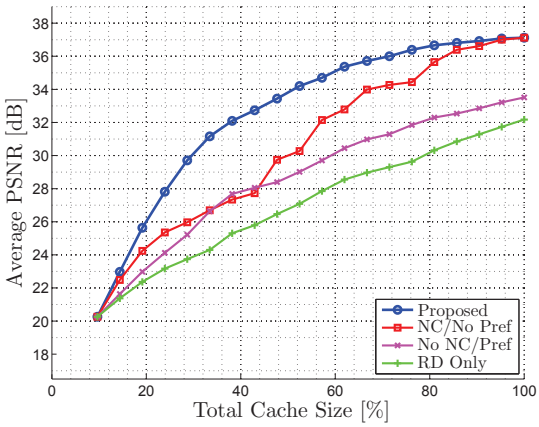
The first thing we notice is that our technique consistently outperforms all the references. The gain is particularly visible when the total cache size is higher than 10 % of the content size and lower than 90 %. All techniques give similar results when a very small cache size is available, since the preferences of the users cannot be taken into account given the constraint imposed by the coding structure, and the techniques using network coding use too small a coding window to gain a substantial advantage over the non-coding techniques.

When the cache size is large –*e.g.*, larger than 90 %– the coding techniques give very similar results as, sending linearly independent combinations of the whole content, they are able to deliver almost all the frames of all the views. The non-coding techniques, on the other hand, suffer from the caching space lost in duplicates, due to the fact that each node stores a decodable stream, which results in a smaller effective caching space.

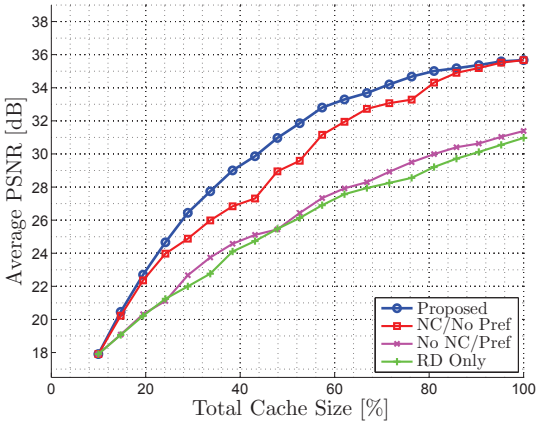
We notice that the highest gain is obtained when the total cache size is around 40 % of the content size, reaching in some cases more than 5 dB. Furthermore, we remark that the different models M1 and M2 only have a minor effect on the performances of the techniques, which present the same trends.



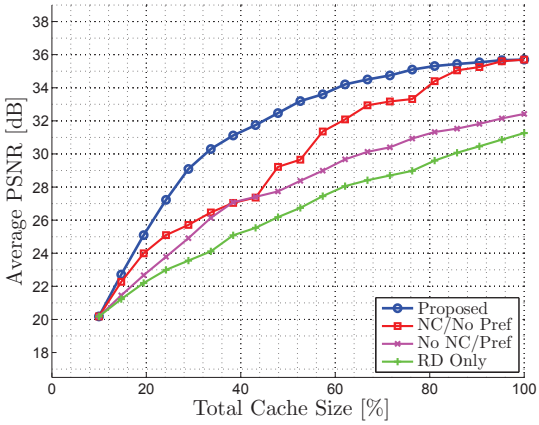
(a) QP=34, $p_c = 0.5$



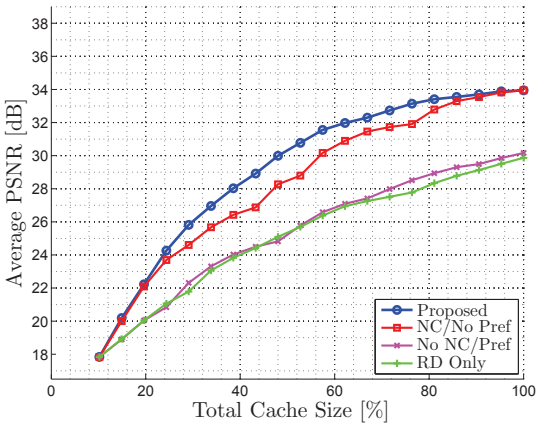
(b) QP=34, $p_c = 0.66$



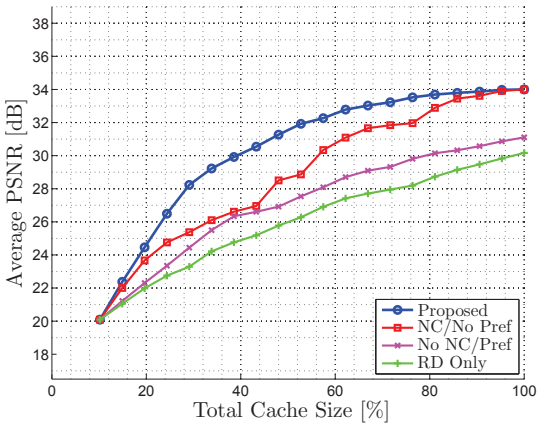
(c) QP=37, $p_c = 0.5$



(d) QP=37, $p_c = 0.66$



(e) QP=40, $p_c = 0.5$



(f) QP=40, $p_c = 0.66$

Figure 4.5: Comparison of the proposed technique with the references for different QPs and values of p_c (for model M2). The average PSNR of all the sequences is plotted against the total cache size expressed as a percentage of the content size.

In Figure 4.5 we present another comparison of the average received video quality for the techniques, varying the QPs (34, 37, and 40) and values of p_c (0.5 and 0.66). Since the two preference models produce very similar results, we present here only the results relative to model M2.

The first thing we notice is that in all the scenarios our technique still outperforms all the references and that, in general, all the techniques rank in the same order as in Figure 4.4.

We also notice that changing the QP used to encode the sequences does not have a great impact on the performances other than a scaling of the quality range, in the sense that the differences among the techniques stay the same proportionally with the quality of the stream itself.

Furthermore, we notice that the gain of the preference-aware techniques, as expected, increases when p_c is larger, *i.e.*, when the distribution of the preferences is not uniform; in the figures an increase from $p_c = 0.5$ to 0.66 results in a gain of about 1 dB.

In order to see how the particular multi-view sequence being transmitted affects the performance of the technique, in Figures 4.6 and 4.7, rather than presenting the results averaged over all the sequences, we show the effect of varying the probability p_c for each sequence individually. These tests are performed with a QP of 31 and using preference model M2.

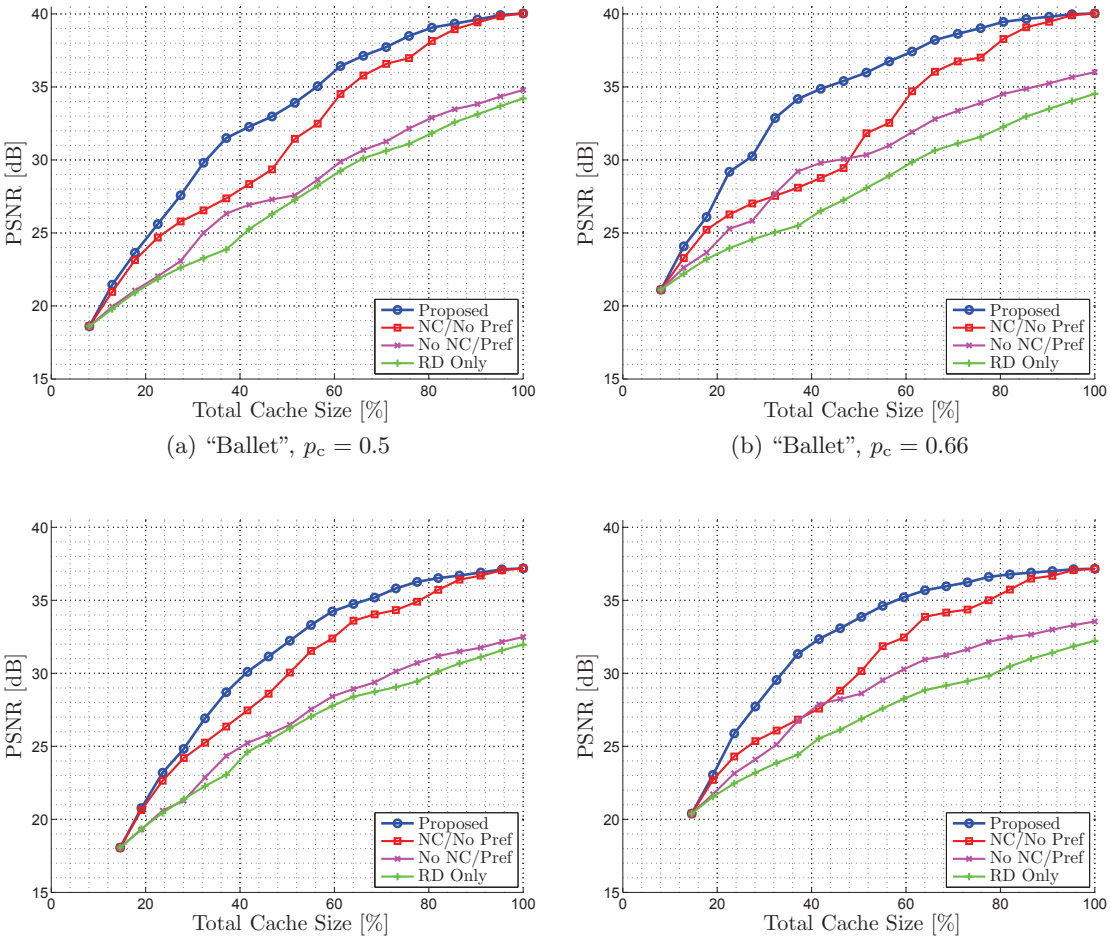
As we can see, each of the sequence has a behavior quite close to the average, in the sense that no sequence is particularly well-behaved or ill-behaved with respect to our technique. Even though non-negligible fluctuations are visible, we ascribe these variations mostly to the quality of the MVC coding process than to the caching.

The effect of a higher preference for the central view is even more visible in Figure 4.8, where we compare, for the average of the sequences, the performances of all the techniques for a fixed QP= 31, for two different larger values of p_c (0.75 and 0.9).

In fact, we can notice that not only our technique gains an even larger advantage over the other techniques, we also notice that the second reference, which is preferences-aware but does not use NC, surpasses –for small sizes of the total cache space– the first, which conversely uses NC but no preferences.

In Fig. 4.9 we report again one of the scenarios in Fig. 4.5, but in this case only 10 nodes are active in the system. We notice that the general trend of the curves remains the same. However, the preference-based techniques perform slightly worse. We attribute this to the fact that the realization of the users preferences, for such a small set of nodes, does not match the model exactly. Also, it should be noted that in order to achieve a large share of the content available in the cache, each individual node has to contribute a larger number of packets. Conversely, we did not observe any significant change when the number of nodes is larger than 100.

Finally, for the sake of completeness, we report a summary of the performance gains over all QPs and over all values of p_c, for both model M1 (in Table 4.3) and M2 (in



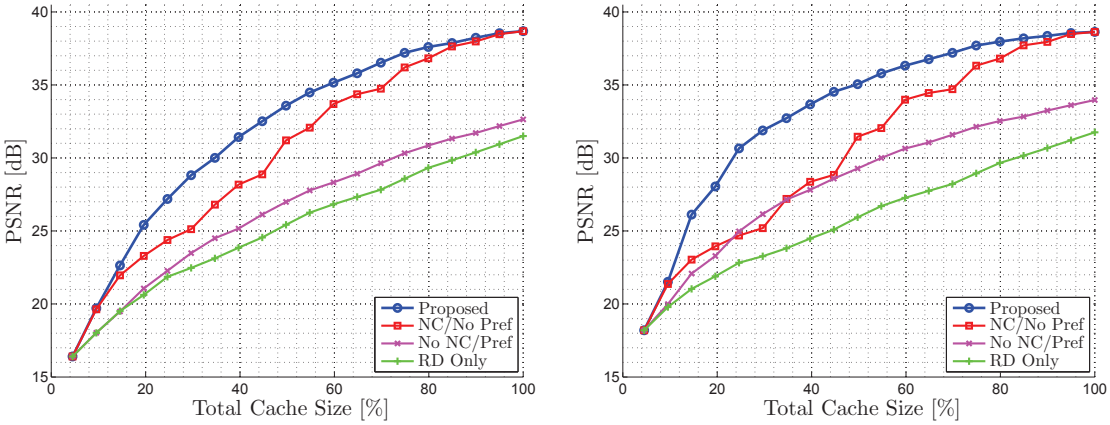
(a) "Ballet", $p_c = 0.5$

(b) "Ballet", $p_c = 0.66$

(c) "Bookarrival", $p_c = 0.5$

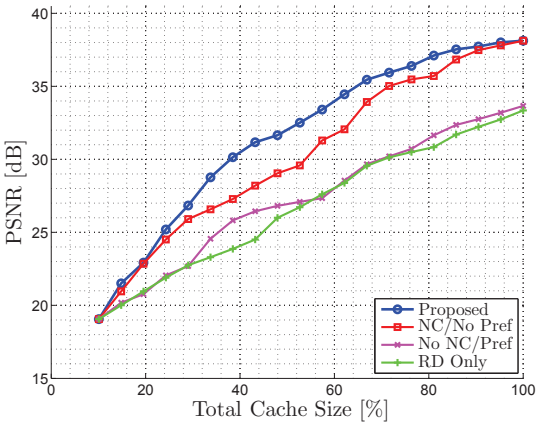
(d) "Bookarrival", $p_c = 0.66$

Figure 4.6: Comparison of the proposed technique with the references for sequences "Ballet" and "Bookarrival" and different values of p_c (for model M2 and QP= 31). The PSNR of each sequence is plotted against the total cache size expressed as a percentage of the content size.

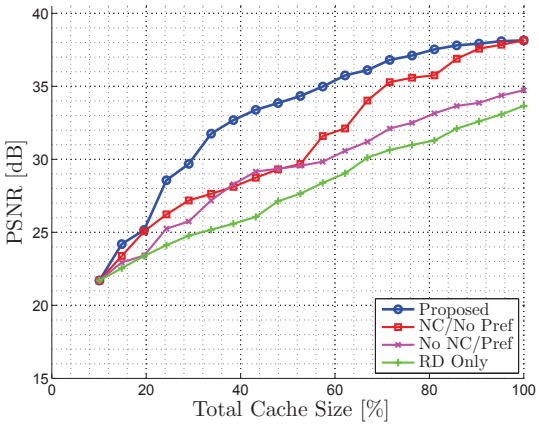


(a) "Breakdancers", $p_c = 0.5$

(b) "Breakdancers", $p_c = 0.66$

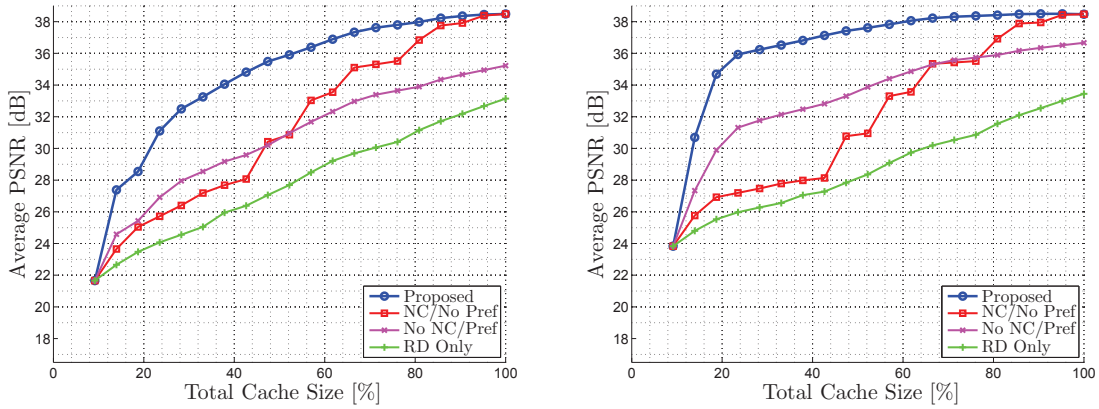


(c) "Doorflowers", $p_c = 0.5$



(d) "Doorflowers", $p_c = 0.66$

Figure 4.7: Comparison of the proposed technique with the references for sequences "Breakdancers" and "Doorflowers" and different values of p_c (for model M2 and QP= 31). The PSNR of each sequence is plotted against the total cache size expressed as a percentage of the content size.



(a) $p_c = 0.75$ (b) $p_c = 0.9$

Figure 4.8: Comparison of the proposed technique with the references for QP= 31 and different values of p_c (for model M2). The average PSNR of all the sequences is plotted against the total cache size expressed as a percentage of the content size.

QP	p_c	Total Cache Size 25 %		Total Cache Size 50 %		Total Cache Size 75 %	
		PSNR [dB]	Δ [dB]	PSNR [dB]	Δ [dB]	PSNR [dB]	Δ [dB]
31	0.50	24.88	+1.31	32.47	+2.27	36.77	+1.13
31	0.66	28.49	+3.45	34.40	+3.92	37.40	+1.65
31	0.75	31.20	+5.33	35.54	+4.90	37.76	+1.94
31	0.90	35.96	+8.70	37.45	+6.53	38.35	+2.43
34	0.50	24.46	+1.16	31.63	+1.91	35.53	+1.00
34	0.66	27.83	+3.08	33.42	+3.43	36.13	+1.47
34	0.75	30.25	+4.68	34.47	+4.32	36.47	+1.74
34	0.90	34.67	+7.74	36.21	+5.80	37.03	+2.19
37	0.50	24.03	+0.96	30.83	+1.63	34.30	+0.90
37	0.66	27.11	+2.63	32.42	+2.95	34.85	+1.32
37	0.75	29.23	+3.95	33.37	+3.75	35.17	+1.56
37	0.90	33.37	+6.78	34.95	+5.08	35.69	+1.95
40	0.50	23.52	+0.73	29.84	+1.34	32.83	+0.79
40	0.66	26.17	+2.03	31.21	+2.47	33.32	+1.15
40	0.75	28.14	+3.23	32.04	+3.16	33.60	+1.35
40	0.90	31.89	+5.72	33.43	+4.31	34.06	+1.68

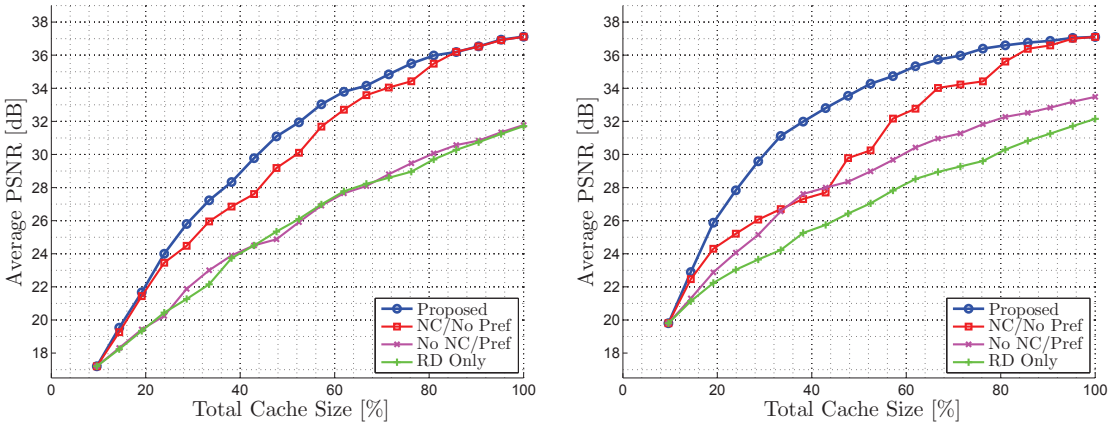
Table 4.3: Summary of average over all sequences of the PSNR gain with respect to a non-preference based network coding technique, for preferences models M1. For each QP we report the average PSNR of the proposed technique and the gain achieved with respect to the reference when the total cache size is 25, 50, and 75 % of the content size.

QP	p_c	Total Cache Size 25 %		Total Cache Size 50 %		Total Cache Size 75 %	
		PSNR [dB]	Δ [dB]	PSNR [dB]	Δ [dB]	PSNR [dB]	Δ [dB]
31	0.50	25.70	+1.26	32.86	+2.52	36.96	+1.23
31	0.66	29.03	+3.39	34.67	+4.09	37.53	+1.72
31	0.75	31.54	+5.23	35.68	+4.97	37.85	+1.99
31	0.90	36.03	+8.60	37.49	+6.55	38.37	+2.43
34	0.50	25.23	+1.08	32.09	+2.29	35.74	+1.14
34	0.66	28.34	+3.01	33.72	+3.66	36.27	+1.57
34	0.75	30.58	+4.59	34.63	+4.43	36.57	+1.81
34	0.90	34.74	+7.64	36.26	+5.82	37.05	+2.20
37	0.50	24.74	+0.84	31.29	+2.03	34.48	+1.04
37	0.66	27.59	+2.54	32.70	+3.19	34.98	+1.42
37	0.75	29.56	+3.86	33.53	+3.88	35.26	+1.63
37	0.90	33.44	+6.67	35.00	+5.11	35.70	+1.96
40	0.50	24.18	+0.60	30.23	+1.67	32.99	+0.92
40	0.66	26.63	+1.95	31.45	+2.67	33.43	+1.23
40	0.75	28.45	+3.16	32.20	+3.28	33.67	+1.41
40	0.90	31.95	+5.62	33.47	+4.34	34.07	+1.68

Table 4.4: Summary of average over all sequences of the PSNR gain with respect to a non-preference based network coding technique, for preferences models M2. For each QP we report the average PSNR of the proposed technique and the gain achieved with respect to the reference when the total cache size is 25, 50, and 75 % of the content size.

Table 4.4). We compare our technique with the non-preference based network coding technique, which in most cases has been shown to outperform the other reference techniques.

In these tables, we report the PSNR of the proposed technique, and the PSNR gain with respect to the non-preference based network coding reference technique, achieved when the total cache size is 25, 50, and 75 % of the content size, for the two models M1 and M2 with various values of the central view probability p_c.



(a) QP=34, $p_c = 0.5$ (b) QP=34, $p_c = 0.66$

Figure 4.9: Comparison of the proposed technique with the references for QP= 34 and for different values of p_c (for model M2). Only 10 nodes are active in the system. The average PSNR of all the sequences is plotted against the total cache size expressed as a percentage of the content size.

4.5 Conclusions

In this chapter, we have presented a novel technique for distributed caching of multi-view video content in a social group. The key idea is to exploit the users' preferences to keep in the distributed cache only the content parts more likely to be requested. We compared the performance of our technique with a network coding technique that does not consider the users' preferences, storing an equal number of frames per view. We observed that the introduction of the preferences, jointly with the constraint imposed on the decodability of the selection, significantly improves the performance with respect to the reference technique, in terms of video quality (PSNR) for a given ratio of the content available in the cache. This technique and the relative results are the object of a journal article currently in preparation.

Possible future work includes the development of a large-scale interactive multi-view distribution system. As opposed to caching the content accessed by the nodes, the system could pre-emptively store part of it in strategic nodes based on predictions of the users' future requests. The predictions could be inferred from the preferences of socially-close nodes. Also, while we illustrated our approach within the MVC framework, it can straightforwardly be applied to multi-view video plus depth [SBB+12].

Chapter 5

Towards Low-Overhead Network Coding with Blind Source Separation

Contents

Blind Source Separation (BSS) is the problem of recovering a set of source signals from a set of observed mixtures, when little or no knowledge of the mixing process is available. Finite field source separation can have an interesting application in the context of network coding, by relieving the nodes from the need to send the combination coefficients, thus largely reducing the overhead cost.

However, the state-of-the art entropy-based methods alone provide an insufficient degree of accuracy to replace practical network coding. In this chapter, after a review of the related work, we present two novel techniques we have recently proposed to increase the discriminating power of the classical entropy-based source separation methods.

The chapter is organized as follows. First, in Section 5.1, we provide an introduction to blind source separation in general, and in particular in the context of finite fields, with an overview of the state of the art separation techniques. In Section 5.2, we present our first contribution in the context of blind source separation, *i.e.*, a technique that improves

the efficiency of the separation algorithm by using non-linear error-detecting coding of the source symbols. In Section 5.3, we present our second contribution, a technique that improves on the previous one by using a message digest to reduce and control the overhead. For both techniques, we present an experimental validation of our proposed approach. Finally, in Section 5.4, we draw our conclusion and outline some current and future work.

5.1 Blind source separation over finite fields

Blind Source Separation (BSS) [CJ10] consists in recovering a set of source signals $\mathbf{S}$ from a set of mixed signals $\mathbf{X} = f(\mathbf{S})$ (also referred to as *observations*) without knowing the sources themselves nor the mixing process parameters. This is a subject that has been intensively investigated in the last three decades, due to its potential numerous applications in fields such as neural networks, speech recognition, sensor signal processing, and biomedical signal processing.

For most practical applications the interesting case is that of *linear* mixing, in which the observations are an unknown linear combination of the source signals:

$$\mathbf{X} = \mathbf{AS}.$$

In this case, the separation process is reduced to finding the combination matrix $\mathbf{A}$ or, equivalently, its inverse $\mathbf{A}^{-1} = \mathbf{W}$.

The *Independent Component Analysis* (ICA) [Com94, HO00] approach solves the BSS problem relying on the assumption that the sources are in the real $\mathbb{R}$ or complex field $\mathbb{C}$, are statistically independent and non-Gaussian. Given a set of observations, ICA algorithms return a set of estimated source signals such that a separation criterion – referred to as *contrast function*– is maximized. Separation criteria can be derived based on information-theoretic principles (*e.g.*, maximizing the entropy or minimizing a Kullback-Leibler divergence) and approximated based on higher-order statistics. Other approaches directly build on higher order statistics. In any case, the assumptions of independence and non-Gaussianity are explicitly used, as most algorithms assume (either directly or indirectly) non-Gaussianity as a measure of regularity. They rely on the fact that, given a linear combination of several *i.i.d.* random variables, the observations will be "more Gaussian" (*e.g.*, in terms of kurtosis), motivated by the Central Limit Theorem, and have a higher entropy than the original sources. Moreover, for sources supposed to be Gaussian only second-order statistics are available.

Whatever *contrast function* be used to discriminate between sources and mixtures, one should note that the original sources can only be retrieved up to some ambiguities. Namely, there will be a permutation ambiguity, *i.e.*, the algorithm will not be able to tell which reconstructed source is which, and scaling ambiguity, *i.e.*, the reconstructed sources will be identified up to a scaling factor. For linear mixing the ambiguities in the reconstructed

sources $\widetilde{\mathbf{S}}$ can be expressed in the form:

$$\widetilde{\mathbf{S}} = \Sigma \cdot \Pi \cdot \mathbf{S}.$$

where Σ is a scaling matrix, *i.e.*, a diagonal matrix of scaling factors, and Π is a permutation matrix.

The problem of ICA has been recently extended to the case of finite fields [Yer07], which presents several additional challenges for ICA, due to the nature of the operations defined over a finite field. In particular, the Central Limit Theorem, which is used in real-valued ICA, does not hold true in a finite field. However, it is still true that a method can be developed, based on the fact that the entropy of any linear combination of statistically independent random variables over GF(q) is larger than the entropy of any of the components, as long as none of the components is uniform. Separation is therefore possible by finding the inverse linear transformation that minimizes the marginal entropy of the resulting combinations. Since the operations take place in a finite field, an exhaustive approach is possible, *i.e.*, to try any possible linear combinations of observations until we find the one that has the lowest entropy [Yer07].

While source separation in finite fields was initially introduced as an interesting theoretical novelty, its potential can be seen for practical applications too. For instance, it has been suggested that BSS schemes over finite fields can be used in the context of eavesdropping over MIMO multi-user digital communications systems [Yer11]. Later in this chapter, we will introduce another application in the context of digital data transmission.

Several algorithms have been proposed to reduce the search space and the execution time of blind source separation algorithms, at the expense of the accuracy [GGT10, GGYT12].

One such technique has been proposed for finite fields of prime order, but can be easily extended to the general case [Yer07]. At each iteration, the algorithm finds a couple of observation vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ and a scalar k in the finite field such that $H(\mathbf{x}_i + k\mathbf{x}_j) < H(\mathbf{x}_i)$ and replaces $H(\mathbf{x}_i)$ with $H(\mathbf{x}_i + k\mathbf{x}_j)$. When no possible substitution can be found, the algorithm terminates, and the final value of the $\mathbf{x}_i$ will be the reconstruction of the original sources. This algorithm is significantly faster than the exhaustive search, but it is prone to local minima.

Other methods to speed up the execution have been proposed, *e.g.*, approximating the entropy with $-p_{\max}\log(p_{\max})$, where $p_{\max}$ is the probability of the most probable element [GGT10, GGYT12].

Since the scope of this chapter is focused on success rate rather than complexity, we shall compare ourself to the *Ascending Minimization of Entropies for ICA* (AMERICA) method [Yer07], originally proposed for GF(2). This method extracts a single source, then removes the contribution from this source to the mixtures and repeats this process N times, after which it has found all N sources, restricting the search space to vectors

linearly independent from the ones recovered so far. Our technique will also follow the same approach, but the search space will be further restricted to vectors that yield admissible sources, *i.e.*, codewords.

In this chapter we focus on improving the results of the separation method by increasing the discriminating power of the algorithm without adding constraints on the distribution of the sources. The rationale is that many of the sources in today's applications do have a distribution close to the uniform, *e.g.*, video sources that have been coded with some form of entropy coding, so the traditional methods fail in this case.

We propose to augment the discriminating power of the ICA methods with different forms of non-linear pre-processing of the sources, towards a more practical application with higher GF orders and sources closer to uniform.

5.2 Error-detecting code based separation algorithm

In this section we present our first contribution to blind source separation in finite fields. The practical interest of this type of technique is that an efficient source separation algorithm over finite fields could be used in the design of a transmission scheme similar to network coding. Since in practical network coding, the random coefficients must be added to the packet as headers, that incurs an overhead that can be of importance if the maximum packet size is small.

On the other hand, in a BSS based approach, it could be possible to relieve the nodes from the need to include the coefficients in the packets, thus reducing significantly the amount of data that has to be transmitted to the receiver in order to decode the packets. Such an approach would instead rely on the capability of the receivers to reconstruct the coefficient themselves.

In this section, we describe our proposed method to separate a number of linearly combined (*mixed*) independent sources defined in a finite field. Generally speaking, the ability of an algorithm to identify a source given a set of mixed observations (*demixing*) stems from the ability to identify a property that holds true for the original sources but does not hold for the mixtures. For instance, entropy based methods assume that the original sources have lower entropy than the mixtures.

Our main idea is to increase the discriminating power of the algorithm by pre-processing the sources with an error-detecting code. Ideally, the code should be such that only the original sources belong to the code, whereas any other possible mixtures do not. This is in practice unfeasible, but we can design a code such that the probability of a mixture accidentally being a codeword is very low. Also, the code cannot be linear, otherwise mixtures would always belong to it; we therefore consider only non-linear codes.

A simple example of non-linear code is the odd-parity bit-code. A parity bit-code is a systematic code consisting in adding a *parity bit* to the source symbol to ensure that the number of bits with the value one in the encoded symbol is always even (even-parity bit-

code) or odd (odd-parity bit-code). Parity bit codes are the simplest form of error detecting code, and have been in use, both in hardware and in software applications, since the 1950s. For our purposes, we use an odd-parity bit-code because it is obviously non-linear, as the null-string is not a codeword (since it has zero bits with value one and zero is an even number). A detailed analysis of the discriminating power of the odd-parity bit-code, *i.e.*, its ability to distinguish between sources and mixtures, is given in the Section 5.2.1.

Let us consider a set of N independent *source signals* $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{N-1}$, each containing T samples, defined in a finite field GF $\left(2^b\right)$, therefore having a length of $T \times b$ bits.

First of all, the sources are encoded with an odd-parity bit-code, such that each element in the *encoded source* $\mathbf{z}_n$ belongs to GF $\left(2^{b+1}\right)$, for a total length of $T(b+1)$ bits, because of the added parity bit, and has an odd number of bits equal to one in its binary representation.

Let us call $\mathbf{Z}$ the $N \times T$ matrix which has $\mathbf{z}_n$ as its n-th row, for $n \in \{0 \ldots N-1\}$. These encoded sources are combined with an unknown mixing $N \times N$ matrix $\mathbf{A}$, also defined in GF $\left(2^{b+1}\right)$. Thus the *observations matrix* will be:

$$\mathbf{X} = \mathbf{A}\mathbf{Z}$$

In order for our separation problem to have a solution, we assume that the matrix $\mathbf{A}$ is invertible, *i.e.*, $\mathbf{rank}(\mathbf{A}) = N$. Each row $\mathbf{x}_n$ of $\mathbf{X}$ is a linear combination, or mixture, of the encoded sources.

In order to recover the original sources, we proceed according to Algorithm 3, as follows. For each vector $\mathbf{w_i}$ of length N in GF $\left(2^{b+1}\right)$, with $i \in \left\{0, \ldots, 2^{N(b+1)}-1\right\}$, we try to *demix* one encoded source:

$$\widetilde{\mathbf{z_i}} = \mathbf{w_i}^\top \mathbf{X}, \quad \forall i \in \left\{0, \ldots, 2^{N(b+1)}-1\right\}.$$

Each vector $\widetilde{\mathbf{z_i}}$ obviously has T elements, if all of them are codewords, we decode the vector, *i.e.*, we remove the parity bit from its elements, thus obtaining the estimated original source $\widetilde{\mathbf{s}}$, and we estimate its entropy $H(\widetilde{\mathbf{s}})$. Notice that the probability of a random mixture being a codeword decreases with T.

After all the vectors $\mathbf{w_i}$ have been tried, we select the N linearly independent vectors corresponding to the demixed sources with the lowest entropy. The matrix $\mathbf{W}$, composed by these vectors, is our estimation of the inverse matrix of $\mathbf{A}$. We limit ourselves to a family of linearly independent vectors under the assumption that $\mathbf{W}$, being the inverse of $\mathbf{A}$, has full rank N. The demixed sources corresponding to this matrix $\widetilde{\mathbf{Z}} = \mathbf{W}^\top \mathbf{X}$ will represent our estimation of the encoded sources. It will suffice to remove the parity bits in order to recover the original sources up to a scaling and permutation ambiguity.

Algorithm 3 Separation algorithm.

Input: $(N \times T)$ mixed sample matrix $\mathbf{X}$.
Output: $(N \times T)$ separated source matrix $\widetilde{\mathbf{S}}$.
$\mathbf{V} \leftarrow \emptyset$, $\mathbf{W} \leftarrow \emptyset$;
for all $\mathbf{w}$ of length N in GF $\left(2^{b+1}\right)$ **do**
  $\widetilde{\mathbf{z}} \leftarrow \mathbf{w}^\top \mathbf{X}$;
  if $\widetilde{\mathbf{z}}$ is a codeword **then**
    $\mathbf{V} \leftarrow \mathbf{V} \cup \{\mathbf{w}\}$;
  end if
end for
repeat
  $\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w} \in \mathbf{V}} \left\{H\left(\mathbf{dec}\left(\mathbf{w}^\top \mathbf{X}\right)\right)\right\}$;
  if $\mathbf{w}^* \notin \text{SPAN}\left(\mathbf{W}\right)$ **then**
    $\mathbf{W} \leftarrow \mathbf{W} \cup \{\mathbf{w}^*\}$;
  end if
  $\mathbf{V} \leftarrow \mathbf{V} - \{\mathbf{w}^*\}$;
until $\|\mathbf{W}\| = N$
$\mathbf{W} \leftarrow$ matrix built from the row vectors in $\mathbf{W}$;
$\widetilde{\mathbf{Z}} \leftarrow \mathbf{W}^\top \mathbf{X}$;
$\widetilde{\mathbf{S}} \leftarrow \mathbf{dec}\left(\widetilde{\mathbf{Z}}\right)$;

5.2.1 Analysis of the discriminating power of odd-parity bit-codes

In this section, we evaluate the probability of a random linear combination of N sources encoded with an odd-parity bit code of being a codeword itself. This probability is useful to assess the augmented discriminating power providing by the encoding with respect to the separation of the sources.

Let $\mathcal{C}$ be the application associating a codeword to each element of GF $\left(2^{b-1}\right)$, or $\mathcal{C}$: GF $\left(2^{b-1}\right) \rightarrow$ GF $\left(2^b\right)$. This simply amounts to add a odd-parity bit to the binary representation of the element. Let $\mathbf{I}_C \subset$ GF $\left(2^b\right)$ be the image of $\mathcal{C}$.

One important property of $\mathcal{C}$ is that, by construction, $0 \notin \mathbf{I}_C$ and $1 \in \mathbf{I}_C$, $\forall b \in \mathbb{N}$. Also, it is easy to see that $\|\mathbf{I}_C\| = \frac{\|\text{GF}\left(2^b\right)\|}{2} = 2^{b-1}$, *i.e.*, half of the elements of GF $\left(2^b\right)$ are codewords. We can therefore infer that, if a value α is drown from a uniform distribution over GF $\left(2^b\right)$, P $\{\alpha \in \mathbf{I}_C\} = \frac{1}{2}$.

Let us consider a monomial $x = \alpha s$, with $\alpha \in$ GF $\left(2^b\right)$ and $s \in \mathbf{I}_C$. In order to evaluate the probability P $\{x \in \mathbf{I}_C\}$, we decompose the sample space in the following way:

$$\begin{aligned} \text{P}\left\{x \in \mathbf{I}_C\right\} &= \text{P}\left\{\alpha s \in \mathbf{I}_C\right\} \\ &= \text{P}\left\{\alpha s \in \mathbf{I}_C \mid \alpha = 0\right\} \text{P}\left\{\alpha = 0\right\} \\ &\quad + \text{P}\left\{\alpha s \in \mathbf{I}_C \mid \alpha = 1\right\} \text{P}\left\{\alpha = 1\right\} \\ &\quad + \text{P}\left\{\alpha s \in \mathbf{I}_C \mid \alpha \neq 0, 1\right\} \text{P}\left\{\alpha \neq 0, 1\right\}. \end{aligned} \tag{5.1}$$

We operate this decomposition on the base of the properties of elements 0 and 1 with

respect to to multiplication: $0 \cdot s = 0 \notin \mathbf{I}_C$ and $1 \cdot s = s \in \mathbf{I}_C$ with probability 1. In the remaining cases, *i.e.*, when $\alpha \neq 0$ and $\alpha \neq 1$, it is easy to verify that the probability of the monomial being a codeword is $\frac{1}{2}$, based on the fact that the product of a scalar other than 0 for all the other elements of the finite amounts to a reordering of the elements. The probability of $\alpha = 0$ (respectively, $\alpha = 1$) being one out of the number of elements in GF $\left(2^b\right)$, we can rewrite Eq. (5.1) as:

$$\begin{aligned} \mathrm{P}\left\{x \in \mathbf{I}_C\right\} &= 0 \cdot \tfrac{1}{2^b} + 1 \cdot \tfrac{1}{2^b} + \tfrac{1}{2} \cdot \tfrac{2^b-2}{2^b} \\ &= \tfrac{1}{2^b} + \tfrac{1}{2} - \tfrac{1}{2^b} \\ &= \tfrac{1}{2}. \end{aligned}$$

The properties of elements zero and one with respect to multiplications become relevant if we consider, instead of the product of two scalars, the product of a scalar by a vector of T elements, *i.e.*, $\mathbf{x} = \alpha\mathbf{s}$ with $\alpha \in$ GF $\left(2^b\right)$ and $\mathbf{s} \in \mathbf{I}_C^T$.

We define *codevector* any vector of GF $\left(2^b\right)^T$ such that each one of its elements is a codeword. In this case we observe that $\forall t \in \{1 \dots T\}$, $1 \cdot s_t \notin \mathbf{I}_C$ and $0 \cdot s_t \in \mathbf{I}_C$. In other words, if $\alpha = 0$ or $\alpha = 1$, the events $\alpha s_t \in \mathbf{I}_C$ for all t are not independent, whereas given any other α, these events are independent with probability $\frac{1}{2}$.

We can therefore operate the same partition as in Eq. (5.1), and write the probability of $\mathbf{x} \in \mathbf{I}_C^T$ as a function of the finite field size 2^b, or equivalently of b, and the vector length T:

$$\begin{aligned} \pi_1(b,T) &\triangleq \mathrm{P}\left\{\mathbf{x} \in \mathbf{I}_C^T\right\} \\ &= \tfrac{1}{2^b} + \left(\tfrac{1}{2}\right)^T \left(1 - \tfrac{2}{2^b}\right) \\ &= 2^{-b} + 2^{-T}\left(1 - 2^{1-b}\right). \end{aligned} \quad . \tag{5.2}$$

The function $\pi_1(b,T)$ is defined as the probability of a single (vector) monomial $\alpha\mathbf{s}$ of being a codevector. Let us now evaluate the probability $\pi_2(b,T)$ that a mixture of two sources is a codevector. Note that all sources are by hypothesis codevectors.

Let $\mathbf{x_2} = \alpha_1\mathbf{s_1} + \alpha_2\mathbf{s_2}$. If we operate an analogous decomposition to that of Eq. (5.2) we obtain:

$$\begin{aligned} \pi_2(b,T) &\triangleq \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T\right\} \\ &= \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T \mid \alpha_1 = 0, \alpha_2 = 0\right\} \mathrm{P}\left\{\alpha_1 = 0, \alpha_2 = 0\right\} \\ &\quad + \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T \mid \alpha_1 = 0, \alpha_2 = 1\right\} \mathrm{P}\left\{\alpha_1 = 0, \alpha_2 = 1\right\} \\ &\quad + \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T \mid \alpha_1 = 1, \alpha_2 = 0\right\} \mathrm{P}\left\{\alpha_1 = 1, \alpha_2 = 0\right\} \\ &\quad + \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T \mid \alpha_1 = 1, \alpha_2 = 1\right\} \mathrm{P}\left\{\alpha_1 = 1, \alpha_2 = 1\right\} \\ &\quad + \mathrm{P}\left\{\mathbf{x_2} \in \mathbf{C}^T \mid \alpha_1, \alpha_2 \neq 0, 1\right\} \mathrm{P}\left\{\alpha_1, \alpha_2 \neq 0, 1\right\} \\ &= 0 \cdot \tfrac{1}{2^{2b}} + 1 \cdot \tfrac{1}{2^{2b}} + 1 \cdot \tfrac{1}{2^{2b}} + 0 \cdot \tfrac{1}{2^{2b}} + \left(\tfrac{1}{2}\right)^T \left(1 - \tfrac{4}{2^{2b}}\right) \\ &= 2^{1-2b} + 2^{-T}\left(1 - 2^{2-2b}\right) \\ &= 2^{2(1-b)-1} + 2^{-T}\left(1 - 2^{2(1-b)}\right). \end{aligned} \tag{5.3}$$

For the case of a linear combination of N sources, let us consider a mixture vector

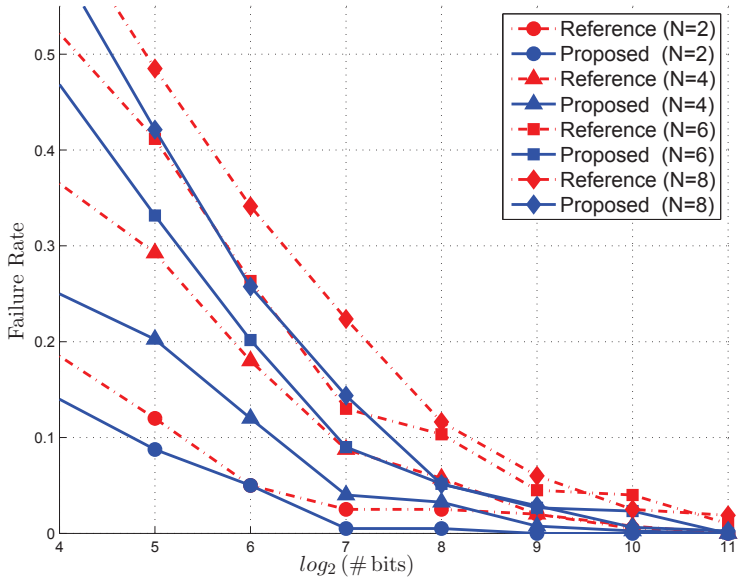


Figure 5.1: Comparison between the reference and the proposed technique for finite field GF(2). The failure rate, *i.e.*, the percentage of sources that the algorithm was not able to identify, is plotted against the number of bits in the mixture in log-scale.

$\mathbf{x_N} = \sum_{n=1}^{N} \alpha_n \mathbf{s}_n$; the expression in Eq. (5.3) can be generalized for N sources as follows:

$$\begin{aligned} \pi_N(b,T) &\triangleq \mathrm{P}\left\{\mathbf{x_N} \in \mathbf{I}_C^T\right\} \\ &= 2^{N(1-b)-1} + 2^{-T}\left(1 - 2^{N(1-b)}\right). \end{aligned}$$

This probability converges to $\pi_N(b) = 2^{N(1-b)-1}$ for $T \to \infty$, therefore we observe that the probability of a random combination of N encoded sources in GF $\left(2^{b-1}\right)$ decreases with the size of the finite field.

This probability can be interpreted as follows: an algorithm based exclusively on the bit code, *i.e.*, identifying any codeword it finds as a source, for sufficiently long sources it would have a rate of false-positive equal of $\pi_N(b)$. Of course our method does not rely solely on the discriminating power of the code, but the coding is used to reduce drastically the search space of the entropy-based method, reducing the running time and improving the success rate.

5.2.2 Experimental results

In the following, we present the results relative to the separation of N sources of T elements for the proposed bit-code based technique, and compare them with the results achievable using an exhaustive entropy-based technique at the same rate.

The reference technique simply consists in identifying the N linear combinations of ob-

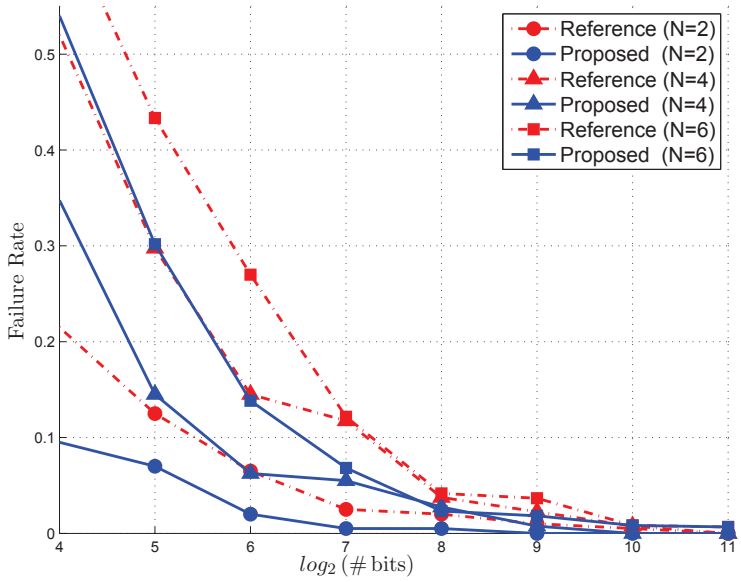


Figure 5.2: Comparison between the reference method and the proposed technique for finite field GF(4). The failure rate is plotted against the number of bits in the mixture in log-scale.

servations such that the combination coefficients are linearly independent and the entropy is minimized [Yer07, GGYT12]. Our technique, on the other hand, is restrained to the linear combinations of observations that yield admissible codewords.

The improvement provided by the augmented discriminating power can be observed in Figs. 5.1 and 5.2 where, for different sizes of the finite field and different number of sources, we report the *failure rate* of the technique *vs.* the number of samples of the observations in log-scale. The failure rate is simply 1 minus the success rate, where the success rate is the number of correctly identified sources divided by the total number of sources. Note that, as mentioned before, a source is considered identified up to a permutation and scaling ambiguity. It is worth noting that, thanks to the properties of the bit-code, even though the scaling ambiguity is still present, it is in practice drastically reduced (see Section 5.2.1 for more details).

We observe that our technique consistently outperforms the reference technique, thanks to the possibility of eliminating candidate solutions with low entropy on the grounds that they are not codewords. Since the failure rate converges to zero with the number of samples for both methods, as we expect, the gain decreases progressively for longer sources. However, the introduction of the non-linear code significantly improves the performances for shorter sources, making the separation viable for relatively shorter signals. Notice that our technique adds an overhead of one bit per symbol to the original sources, however, this overhead is taken into account in the comparisons.

We also report, in Fig. 5.3, a comparison of the two techniques with fixed number of sources ($N = 2$) and fixed number of samples ($T = 256$) to observe how the performances

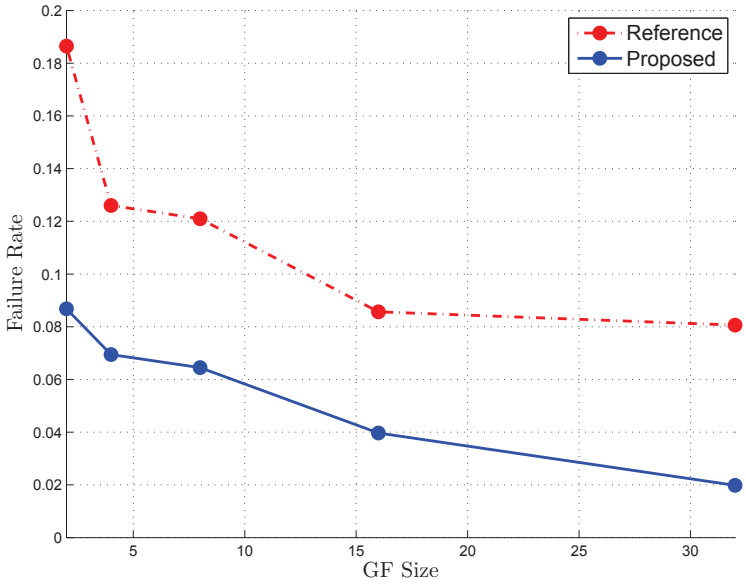


Figure 5.3: Comparison between the reference method and the proposed technique for a fixed number of sources ($N = 2$) and same number of samples ($T = 256$). The failure rate is plotted against the size of the finite field.

GF	8	16	32	64	128	256
$T = 256$	0.29(1.52)	0.59(3.00)	0.89(4.54)	0.89(4.57)	1.09(5.53)	1.29(6.95)
$T = 1024$	1.69(8.50)	1.89(8.64)	1.89(9.55)	1.99(9.64)	1.99(10.01)	1.99(10.10)
$T = 4096$	1.99(8.61)	1.99(9.60)	1.99(9.64)	1.99(10.10)	1.99(10.11)	1.99(10.16)

Table 5.1: Reduction of the failure rate in percentage points when Reed-Solomon codes are used to increase the discriminating power, for two sources and p_1=0.55. Numbers in parenthesis refer to the failure rate when the scaling ambiguity is not tolerated.

of the two methods vary with respect to the size of the finite field. We observe that both techniques perform better when they operate within a larger finite field, but the failure rate is divided by a factor of approximately two for small sizes of the finite field (GF(4)), and by a factor of approximately four for large fields (GF(32)).

5.3 Hashing Based Separation Algorithm

While the results presented in the previous section showed how the entropy based methods benefit from error detecting coding by applying the estimation of the entropy only to solutions that are admissible, our preliminary studies also show that similar results can be achieved with more efficient codes than parity codes.

In particular, we have tested Hamming codes and Reed-Solomon codes, rendered non-linear by complementing the redundant part. Some results for Reed-Solomon codes for different sizes of the finite field and different lengths of the sources are given in Table 5.1. However, the fixed structure of these codes implies a fixed –and non-negligible– amount

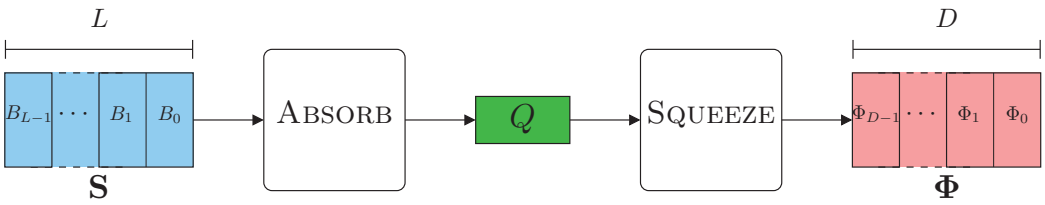


Figure 5.4: Sponge construction of the hashing function. The Absorb function processes an input **S**, divided in L blocks of fixed size, and produces a fixed-length state Q. The Squeeze function can use the state Q to generate a digest Φ of assigned length D.

of overhead.

In this section, we propose a more flexible framework, able to control the amount of overhead introduced with the pre-processing, thus allowing the user to strike the trade-off best suited for the specific scenario.

In order to do so, rather than encoding each symbol of the sources with a pre-defined error-detecting code, we apply a *hashing function* to the sources to generate a variable-length *message digest*.

A hashing function is an algorithm that maps large data sets of variable length into smaller sets. The input of a hashing function is referred to as *message*, whereas its output is referred to as *digest*. These functions are designed so that they are easy to compute, and so that it is unfeasible to generate a message with a given digest, or to modify a message without changing its digest, or to find two different messages having the same digest [BDPVA11].

In our context, rather than a malicious agent or a bit error probability, our hashing function has to be robust with respect to linear combinations. In other words, the digest of a combination of sources should not be equal to the same combination of their digests, *i.e.*, for two sources $\mathbf{x}$ and $\mathbf{y}$ and a hashing function φ:

$$\varphi(\alpha\mathbf{x} + \beta\mathbf{y}) \neq \alpha\varphi(\mathbf{x}) + \beta\varphi(\mathbf{y}). \tag{5.4}$$

This allows the digest values to be used to distinguish between the linear combinations and the original sources.

However, note that since the field is finite, it is impossible to design a function φ for which the non-linearity property expressed in Eq. 5.4 is satisfied for all α, β, $\mathbf{x}$, and $\mathbf{y}$.

A discriminating hashing function will therefore present no false negative (because the function is deterministic), but it will also always return false positives with a probability depending on the design of the function itself.

The correct message digest will therefore point to a set of candidates for the original sources, much smaller than the original search space, on which other criteria, like entropy minimization, can be applied.

Algorithm 4 Absorb part of the sponge construction of the hashing function. Given an input of arbitrary length **S**, it produces a state Q of fixed length.

function Q =Absorb(**S**)
  S is divided into L blocks B_i of 32 bits;
  $\sigma \leftarrow 0$; $K \leftarrow$ 0x99999999; $Q \leftarrow 0$;
  for $i \leftarrow 1$ to L **do**
    $Q \leftarrow [Q \oplus B_i]_r$;
    $Q \leftarrow [Q \oplus \sigma]_r$;
    $Q \leftarrow [Q \oplus K]_r$;
    $\sigma \leftarrow Q$;
    $Q \leftarrow Q \oplus [Q]_r$;
  end for
  return Q
end function

Algorithm 5 Squeeze part of the sponge construction of the hashing function. Given a state Q of fixed length it produces a message digest **Φ** of assigned length D.

function **Φ** =Squeeze(Q, D)
  $\sigma \leftarrow 0$; $K \leftarrow$ 0x99999999;
  for $i \leftarrow 1$ to D **do**
    $Q \leftarrow [Q]_r$;
    $Q \leftarrow [Q \oplus \sigma]_r$;
    $Q \leftarrow [Q \oplus K]_r$;
    $\sigma \leftarrow Q$;
    $Q \leftarrow Q \oplus [Q]_r$;
    $\Phi_i \leftarrow Q$;
  end for
  return **Φ**
end function

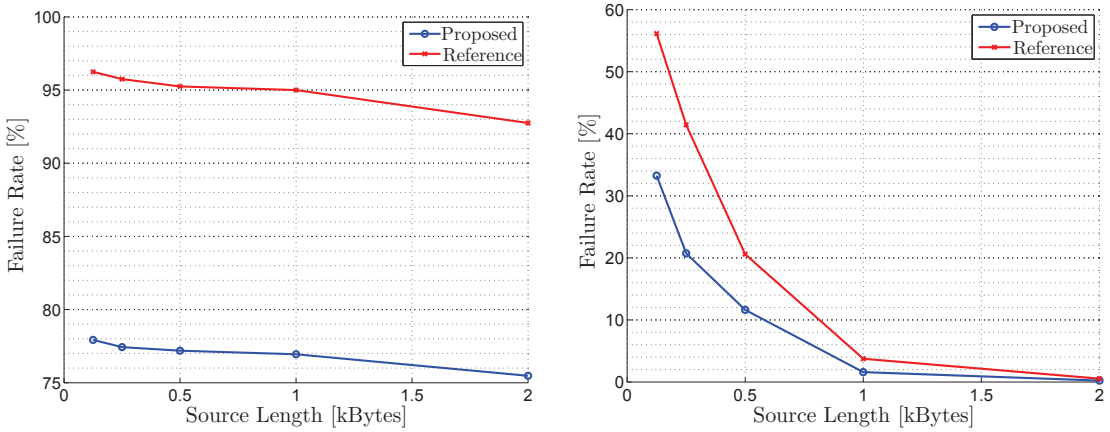
For our separation purposes, we propose to use a *sponge construction* of the hashing function. A sponge construction is a hashing function design technique that allows to decouple the input length and the output length of the hashing function.

Two primitive functions are provided: first an Absorb function that takes a variable-length input **S** and produces a fixed-length state Q, then a Squeeze function that takes the state Q and returns an output **Φ** of arbitrary length D specified by the user [BDPVA11].

In order to process a variable length input, the Absorb function works on blocks of data of fixed length (in our implementation, 32 bits). The input data might need to be zero-padded to fit into an integer number L of blocks.

Our implementation of the Absorb and Squeeze functions is given in Algorithm 4 and 5, respectively. Note that the functions perform the same basic operations, with different inputs and outputs. These functions use basic bit operations commonly used in hashing: modulo-2 sum (*i.e.*, exclusive or) and circular shift.

In the Absorb function, the state Q is initialized to zero. Then, for each iteration i,



(a) Comparison for N=2 sources. The source bits are one with probability p_1=0.5. The overhead is D=1 symbol per source.

(b) Comparison for N=4. The source bits are one with probability p_1=0.52. The overhead is $D = 2$ symbols per source.

Figure 5.5: Comparison between the reference entropy-based method and the proposed digest-enhanced technique, for in GF(2). The failure rate, *i.e.*, the percentage of sources that the algorithm was *not* able to identify, is plotted against the source length (in kilobytes)

one of the L blocks B_i of the input is added to the current state in modulo 2. The result is then circularly shifted of one position (circular shift is denoted in Algorithm 4 and 5 with $[\cdot]_{\mathrm{r}}$).

The same operation of update of the state (*i.e.*, sum and circular shift) are then applied using the value state at the previous iteration σ, and a constant value K. The constant value is chosen to prevent that a long run of zeros in the input might permanently force the state to zero. Finally the state is added to the shifted version of itself.

In the Squeeze function, the operations are the same, except that the blocks B_i are replaced with constant zero blocks, while the output $\mathbf{\Phi}$ is composed of the state Q at the end of each iteration i. The number D of iterations, equal to the number of output symbols, is specified by the user.

These functions work on blocks of fixed size of 32 bits, therefore, in order to produce outputs in fields smaller than $GF\left(2^{32}\right)$, only the first b bits of each symbol Φ_i are considered.

Notice that these computations are easy, and can be implemented efficiently at low level.

5.3.1 Experimental results

In the following, we present the results for the separation of N sources defined in a finite field GF(2^q), for the proposed digest-enhance technique, and compare them with the results achievable using an exhaustive entropy-based technique without overhead, such as described in Sec 5.1.

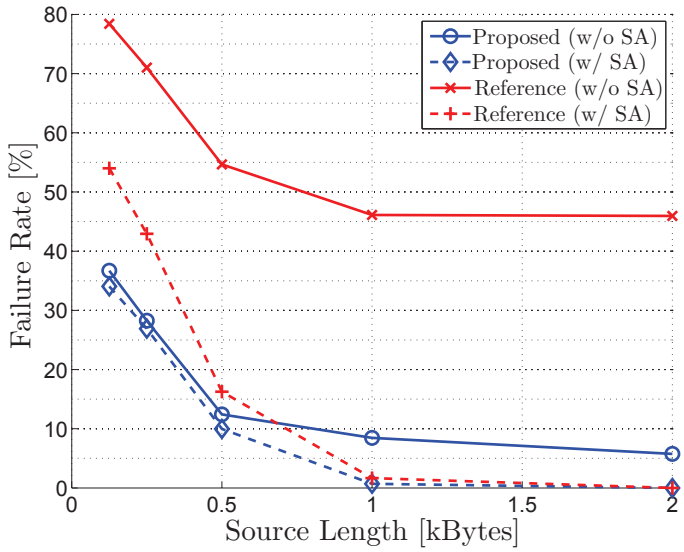


Figure 5.6: Comparison between the reference method and the proposed technique, for $N = 4$ sources in GF(4). The source bits are one with probability p_1=0.52. The overhead is 2 symbols per source. The failure rate is plotted against the source length (in kilobytes). The dashed lines represent the failure rate when the sources are considered identified up to a scaling factor. The solid lines represent the failure rate when scaling ambiguity is not tolerated.

In particular, in our experimental setup, the reference technique simply consists in identifying the N linear combinations of observations such that the combination coefficients are linearly independent and the entropy is minimized [Yer07, GGYT12]. This technique does not alter the sources and does not add any overhead.

Our technique, on the other hand, is restrained to the linear combinations of observations that carry a valid digest, *i.e.*, such that the digest appended to the source is equal to the one locally computed by the separation algorithm.

In order to have a consistent parameter for comparison over different finite fields, the probability distributions of the sources are expressed in terms of p_1, *i.e.*, the probability that a bit is 1. For finite fields larger than GF(2), this probability is applied independently on each bit.

We report in Fig. 5.5(a) the *failure rate* of the technique *vs.* the length of the sources for the case of $N = 2$ sources in GF(2) with uniform distribution (the bit probability $p_1 = 0.5$). The source length includes, for the proposed technique, the overhead —which is in any case of a few bits over several hundreds of bytes and therefore does not affect the figure. Each plotted point corresponds to the average over at least 100 runs of the algorithm, each with randomly generated sources and mixing matrix. The failure rate is simply one minus the success rate, where the success rate is the number of correctly identified sources divided by the total number of sources.

We observe that in the case where the ICA methods have the worst performance

(*i.e.*, uniform distribution), even with just one bit of overhead per source, our technique consistently outperforms the reference method, and the separation is greatly improved for all source lengths.

In Fig. 5.5(b), we also report the results obtained for a higher number of sources ($N = 4$) in the same field GF(2), with an overhead $D = 2$ symbols, in this case two bits.

In this case, where the sources are not uniform ($p_1 = 0.52$), both methods converge to complete separation with the length of the sources. However, our technique still consistently outperforms the reference, an effect more noticeable when the length of the sources is small and the failure rate is reduced by almost a factor two. This result is very important for practical applications, in which the separation is done packet-wise, since packets typically have a size limit dictated by the network.

Notice that, in practical applications, finite fields of order higher than two are typically used, as the probability of randomly generating a mixing matrix that has full rank (and is thus invertible) increases with the size of the field.

In this respect, we present in Fig. 5.6 the results obtained if we consider the same scenario in terms of number of sources and source distribution, but with sources defined in GF(4).

As mentioned in Sec. 5.1, entropy-based methods can only identify sources up to a scaling factor, a limitation known as *scaling ambiguity*. If we tolerate the scaling ambiguity, we observe that the performances for both methods are similar to the previous case, with a failure rate of about 55 % for the reference technique and 35 % for the proposed for sources of about 128 bytes, and a failure rate of less than 1 % for both techniques at about 1 kilobyte.

However, unlike the case of analog applications, scaling ambiguity is often not tolerable in digital applications, *e.g.*, a multiple in finite field of an encoded video packet bares no meaning, and the scaled signal is not semantically equivalent to the unscaled one.

Therefore, if we consider that failure rate in a stricter sense, where we do not tolerate the scaling ambiguity, we see that our technique presents a much lower failure rate than the reference even for longer sources. In fact, without scaling ambiguity, the failure rate of the reference technique increases to 80, % for sources of about 128 bytes, while it remains almost unaltered for the proposed. For sources of about 1 kilobyte, the failure rate of the reference technique is about 50 %, while it stays lower than 10, % for the proposed. Furthermore, while the failure rate of the proposed technique keeps decreasing when the length of the sources increases up to 2 kilobytes the reference technique stays almost flat at 55 %.

5.4 Conclusions

In this chapter we presented our contributions to increasing the discriminating power of blind source separation methods for linear mixtures in a finite field.

In our first contribution, we proposed to use a non-linear channel encoding of the source signals, and in particular, we used an odd-parity bit code, which has the advantage of being very simple to implement. However, these results can be extended to a more general case of a non-linear error detecting code.

The discriminating power is augmented in the sense that the entropy based method will be assisted by the error detecting coding, restraining the estimation of the entropy to the solutions that are admissible in the sense that the reconstructed source is a codeword. This eliminates several solutions that, even if they present low entropy and could be mistakenly identified as sources by the reference technique, cannot be admitted as they are not part of the code.

Our experimental results show that the proposed technique consistently outperforms the reference method, especially in the case of sources with a small number of available samples, which is more critical for the entropy-based methods, making the blind source separation more suitable for practical applications, where the number of samples is typically limited by the size of a packet. This technique has been presented at the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013).

In our second contribution we proposed to generate, for each source, a non-linear and flexible message digest to be sent along the sources. The message digest is generated by a hashing function defined through a sponge-construction, which allows to decouple the input and the output length. In other words, the function is able to generate a digest of any given length for sources of arbitrary length. The message digest is defined to be robust with respect to linear combinations, *i.e.*, a linear combination of digests has very low probability of being equal to the digest of the linear combination of the corresponding messages.

This property is exploited at the receiver side where observations with an invalid digest can be discarded without further processing. On the remaining observations, which are a considerable smaller subset of the search space, traditional entropy-based methods can be applied.

Our results show that this approach dramatically improves the separation ability of the ICA techniques in finite fields, in cases where the traditional approaches are unfeasible, *i.e.*, for short sources with distributions close to uniform. Furthermore, our technique is much more robust to the scaling ambiguity problem, which we argue is much more problematic in digital applications than it is in traditional analog blind source separation.

The possibility of separating efficiently the mixed sources given a small and controllable overhead open the possibility for a transmission scheme similar to network coding, where sources are linearly combined in order to increase the throughput and the loss immunity, but with a significantly reduced overhead.

Conclusion and Perspectives

We believe that network coding is bound to replace or at least complement traditional routing in the next few years. Nowadays, with the ubiquitous presence of social-centered multimedia applications, the multicast model is at least as present as the unicast over the Internet. As we already discussed in this thesis, NC is superior to traditional routing when it comes to multicast, both in terms of throughput and delay. This, jointly with the fact that users' demand for advanced and high quality multimedia services increases every day, motivated our research. The main goal of this thesis has been to cope with the challenges arising from the design of video service-aware network coding techniques in different contexts and environments. In this field, after an in-depth study of the theoretical aspects, we proposed a number of original contributions, most of which already led to international publications, that we summarize in the following.

Video streaming protocol for wireless networks

We addressed the topic of video streaming over wireless ad-hoc networks. We proposed an innovative solution acting both on source and channel encoding by using a joint framework of network coding and multiple description coding. We integrated this framework in a video-delivery protocol able to construct and maintain an overlay network and to keep track of the topology changes with a low overhead in terms of control packets. The protocol performs a distributed online optimization in order to maximize the expected video quality received by the users.

Per-hop transmission scheme of MDC content over wireless network

While with the first proposed technique we focused on the organization of the nodes in an overlay network, with our second contribution we focused on the transmission on a single link. Our goal was to optimize the transmission scheme in terms of loss tolerance and delay. In particular, we argued that expanding window network coding techniques are best suited for real-time streaming, as they allow, with mild hypotheses, an instantaneous decoding. We proposed a novel technique that organizes the frames of the MDC-coded stream in a bi-dimensional frame buffer, and identifies clusters of frames with similar rate and distortion properties. This organization allows us to provide an optimal scheduling order for the

clusters, in the sense that, following the order of inclusion in the coding window provided by the scheduling, at each sending opportunity the expected marginal distortion of the receivers is minimized. Within the optimal clusters, we proposed a randomized choice of the frames in order to allow for diversity, thus increasing the average innovativity and maximizing the throughput of the receivers.

Per-hop transmission scheme for multi-view content

Realizing that the promising results of the previous technique made it interesting in fields other than multiple description coding, we extended it to the case of multi-view video streaming. Extension to the multi-view streaming scenario comes with its own set of challenges; in particular, we had to take into account the complex prediction structure of the multi-view stream and its rate-distortion properties. Also, while the multi-view scheduling requires similar clustering and scheduling procedures, due to the different prediction structures –that include inter-view prediction– a different buffering strategy had to be devised.

Distributed caching system for multi-view content

We proposed a framework to handle an interactive multi-view distributed caching service for users belonging to a social group. The framework is able to provide a high expected video quality requiring that only a small part of the content is stored by the members of the social group. This result is achieved by jointly taking into account the preferences of the users, in terms of preferred point-of-view of the users and the rate-distortion characteristics of the multi-view stream.

Low overhead network coding strategies

We investigated the potential use of blind source separation techniques in finite fields to overcome the overhead problems of network coding, which requires that the coding coefficients are sent along the packets. In particular, we proposed to pre-process the source signals with a non-linear error-detecting code. Used in addition to traditional entropy methods, the pre-processing phase largely improves the discriminating power, *i.e.*, the ability of the algorithm to retrieve the original sources without sending the combination coefficients. Based on the success of this first contribution, we extended the concept of non-linear pre-processing into a more general framework. We proposed a digest-based pre-processing, *i.e.*, to prepend the source signal with a non-linear message digest. This technique, with a low and controllable overhead, provides a much higher separation rate than the traditional entropy-based methods. We proposed a sponge construction of the hashing function used to generate the message digest. The sponge construction allows to generate an arbitrary size digest for a variable size source, with a simple combination of

computationally easy low-level operations (exclusive-or, register shift, *etc.*). The digest-based technique is especially effective if no scaling ambiguity is tolerated, such as the case of multimedia streaming.

Perspectives & Future Work

The results that we have obtained and that we have presented and analyzed throughout this thesis suggest several possible directions for future research that may lead to interesting new innovations.

A first promising axis of research concerns the design of an overlay construction and maintenance protocol for MDC and MVC streaming over mobile networks that maximizes the users' received video quality and minimizes the network load and transmission delay. This protocol should integrate in its optimization model both the rate-distortion characteristics of the source coding and the combination parameters of network coding, *i.e.*, able to construct a logical multi-tree topology that interconnects the nodes of the mobile network in a way that allows network coding to fully exploit its property of maximizing the information flow by jointly taking into account the contribution of each data packet to the overall distortion.

Another possible field of future research is the design of a complete multi-view distribution system that assists every step of the streaming service, from the source coding, to the network coding transmission over every hop, to the distributed replication, *etc.*

Finally, in the context of finite field blind source separation for reduction of network coding overhead, a viable direction of investigation is the definition of an optimality criterion for pre-processing schemes in the sense of an a priori evaluation of the increased separability that the pre-processing is able to provide to a source separation algorithm at the receiver. This criterion should also keep into account the overhead added to the source by the pre-processing, thus enabling a proper rate-constrained optimization.

Publications

Journal articles

1. Claudio Greco, Irina-Delia Nemoianu, Marco Cagnazzo, and Béatrice Pesquet-Popescu, "A Rate-Distortion Optimized Distributed Social Caching System for Interactive Multi-View Streaming using Network Coding" (in preparation).

2. Irina-Delia Nemoianu, Claudio Greco, Marco Cagnazzo, and Béatrice Pesquet-Popescu, "On a Hashing-Based Enhancement of Source Separation Algorithms over Finite Fields for Network Coding Applications" (in preparation).

Conference papers

1. Irina-Delia Nemoianu, Claudio Greco, Marco Cagnazzo, and Béatrice Pesquet-Popescu, "A framework for joint multiple description coding and network coding over wireless ad-hoc networks", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, March 2012.

2. Claudio Greco, Irina Delia Nemoianu, Marco Cagnazzo, and Béatrice Pesquet-Popescu, "A network coding scheduling for multiple description video streaming over wireless networks", *Proceedings of the European Signal Processing Conference*, Bucharest, Romania, August 2012.

3. Irina-Delia Nemoianu, Claudio Greco, Marco Cagnazzo, Béatrice Pesquet-Popescu, "Multi-View Video Streaming over Wireless Networks with RD-Optimized Scheduling of Network Coded Packets", *Proceedings of Visual Communications and Image Processing*, San Diego, CA, USA, November 2012.

4. Irina-Delia Nemoianu, Claudio Greco, Marc Castella, Béatrice Pesquet-Popescu, Marco Cagnazzo, "On a practical approach to source separation over finite fields for network coding applications", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, May 2013.

Book Chapters

1. Irina-Delia Nemoianu, Béatrice Pesquet-Popescu, "Network Coding for Multimedia Communications", in *Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools*, IGI Global, 2013.

2. Claudio Greco, Irina-Delia Nemoianu, Marco Cagnazzo, Jean Lefeuvre, Frédéric Dufaux, Béatrice Pesquet-Popescu, "Multimedia Streaming", in *Elsevier E-Reference Signal Processing* (to appear).

Talks & Presentations

1. Project SWAN, "Source aware network coding", Project presentation at the Digiteo Annual Forum (Poster 51), October 2011.

2. "Codage réseau pour la diffusion de contenus vidéo de haute qualité", Poster at *Journée Futur & Ruptures*, Fondation Télécom, January 2012.

3. "Codage réseau pour la diffusion de contenus vidéo de haute qualité", Presentation at *Journée Futur & Ruptures*, Fondation Télécom, January 2013.

Bibliography

[ACLY00] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow". *IEEE Transactions on Information Theory*, vol. 46 (4), pp. 1204–1216, Jul. 2000. *Cited in Sec.* (document), 1.1, 1.1.1

[AML+05] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu, "Influences on cooperation in BitTorrent communities", in *Proceedings of ACM/SigComm Workshop on Economics of Peer-to-Peer Systems*, Philadelphia, PA, USA, Aug. 2005. *Cited in Sec.* 4.1

[Apo01] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity", in *Proceedings of SPIE International Symposium on Visual Communications and Image Processing*, 2001. *Cited in Sec.* 2.1

[BCF+99] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications", in *Proceedings of IEEE International Conference on Computer Communications*, vol. 1, pp. 126–134, New York, NY, USA, Mar. 1999. *Cited in Sec.* 4.1

[BDPVA11] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The Keccak SHA-3 submission". *Submission to NIST (Round 3)*, 2011. *Cited in Sec.* 5.3

[BM08] J. Bondy and U. Murty, *Graph theory*, Springer, 2008. *Cited in Sec.* 1.1.2

[CCM08] O. Campana, R. Contiero, and G. Mian, "An H.264/AVC video coder based on a multiple description scalar quantizer". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18 (2), pp. 268–272, Feb. 2008. *Cited in Sec.* 2.1

[CFL00] M. Caramma, M. Fumagalli, and R. Lancini, "Polyphase down-sampling multiple-description coding for IP transmission", in *Proceedings of SPIE*, 2000. *Cited in Sec.* 2.3.1

[CJ10] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, first edn., 2010. *Cited in Sec.* 5.1

[COC11] G. Cheung, A. Ortega, and N.-M. Cheung, "Interactive streaming of stored multiview video using redundant frame structures". *IEEE Transactions on Image Processing*, vol. 20 (3), pp. 744–761, Mar. 2011. *Cited in Sec.* 4.1

[Com94] P. Comon, "Independent component analysis, a new concept?" *Signal Processing (Elsevier Science)*, vol. 36 (3), pp. 287–314, Apr. 1994. *Cited in Sec.* 5.1

[CWJ03] P. Chou, Y. Wu, and K. Jain, "Practical network coding", in *Proceedings of Allerton Conference on Communication Control and Computing*, Monticello, IL, USA, Oct. 2003. *Cited in Sec.* (document), 1.1.4, 2.2, 2.2.1

[DGW+10] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems". *IEEE Transactions on Information Theory*, vol. 56 (9), pp. 4539–4551, 2010. *Cited in Sec.* 1.2.1

[EFLBS07] A. El Fawal, J. Le Boudec, and K. Salamatian, "Multi-hop broadcast from theory to reality: practical design for ad-hoc networks", in *Proceedings of IEEE/ACM International Conference on Autonomic Computing and Communication Systems*, Rome, Italy, Oct. 2007. *Cited in Sec.* 2.2

[FCF10] A. Fiandrotti, J. Chakareski, and P. Frossard, "Popularity-aware rate allocation in multiview video", in *Proceedings of SPIE*, vol. 7744, 2010, Invited Paper. *Cited in Sec.* 4.2

[FF56] L. Ford and D. Fulkerson, "Maximal flow through a network". *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956. *Cited in Sec.* 1.1.2, 1.1.3

[FFLT05] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over IP". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15 (3), pp. 321–334, Mar. 2005. *Cited in Sec.* 2.1, 2.3.1

[FHPK13] F. Fitzek, J. Heide, M. Pedersen, and M. Katz, "Implementation of network coding for social mobile clouds". *IEEE Signal Processing Magazine*, vol. 30 (1), pp. 159–164, Jan. 2013. *Cited in Sec.* 4.1

[FJL00] M. Frodigh, P. Johansson, and P. Larsson, "Wireless ad-hoc networking: the art of networking without a network". *Ericsson Review*, vol. 4, pp. 248–263, 2000. *Cited in Sec.* 1.3, 4.1

[Gab85] E. M. Gabidulin, "Theory of codes with maximum rank distance". *Problemy Peredachi Informatsii (Russian Academy of Sciences)*, vol. 21 (1), pp. 3–16, 1985. *Cited in Sec.* 1.5, 1.5.1

[GC11] C. Greco and M. Cagnazzo, "A cross-layer protocol for cooperative content delivery over mobile ad-hoc networks". *Inderscience International Journal of Communication Networks and Distributed Systems*, vol. 7 (1–2), pp. 49–63, Jun. 2011. *Cited in Sec.* 2.2, 2.2.1, 2.3.1

[GCPP10] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "H.264-based multiple description coding using motion compensated temporal interpolation", in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Saint-Malo, France, Oct. 2010. *Cited in Sec.* 2.1

[GCPP12] ———, "Low-latency video streaming with congestion control in mobile ad-hoc networks". *IEEE Transactions on Multimedia*, vol. 14 (4), p. 14pp, 2012, to appear. *Cited in Sec.* 2.2

[GDMC11] N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Wireless video content delivery through distributed caching and peer-to-peer gossiping", in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, pp. 1177–1180, Pacific Grove, CA, USA, Nov. 2011. *Cited in Sec.* 4.2

[GGP01] T. Guionnet, C. Guillemot, and S. Pateux, "Embedded multiple description coding for progressive image transmission over unreliable channels", in *Proceedings of IEEE Internantional Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001. *Cited in Sec.* 2.1

[GGT10] H. Gutch, P. Gruber, and F. Theis, "ICA over finite fields", in *Proceedings of Springer-Verlag International Conference on Latent Variable Analysis and Signal Separation*, pp. 645–652, 2010. *Cited in Sec.* 5.1

[GGYT12] H. W. Gutch, P. Gruber, A. Yeredor, and F. J. Theis, "ICA over finite field—separability and algorithms". *Signal Processing (Elsevier Science)*, vol. 92 (8), pp. 1796–1808, 2012. *Cited in Sec.* 5.1, 5.2.2, 5.3.1

[GK98] V. Goyal and J. Kovačević, "Optimal multiple description transform coding of Gaussian vectors", in *Proceedings of Data Compression Conference*, Snowbird, UT, USA, Mar. 1998. *Cited in Sec.* 2.1

[GK01] V. K. Goyal and J. Kovačević, "Generalized multiple description coding with correlating transforms". *IEEE Transactions on Information Theory*, vol. 47 (6), pp. 2199–2224, Sep. 2001. *Cited in Sec.* 2.1

[GKAV98] V. Goyal, J. Kovačević, R. Arean, and M. Vetterli, "Multiple description transform coding of images", in *Proceedings of IEEE Internantional Conference on Image Processing*, Chicago, IL, USA, Oct. 1998. *Cited in Sec.* 2.1

[GKV99] V. Goyal, J. Kovačević, and M. Vetterli, "Quantized frame expansions as source channel codes for erasure channels", in *Proceedings of Data Compression Conference*, Snowbird, UT, USA, Mar. 1999. *Cited in Sec.* 2.1

[Goy01] V. K. Goyal, "Multiple description coding: compression meets the network". *IEEE Signal Processing Magazine*, vol. 18 (5), pp. 74–93, Sep. 2001. *Cited in Sec.* (document), 2.1

[GPCPP11] C. Greco, G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, "An MDC-based video streaming architecture for mobile networks", in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Hangzhou, PRC, Oct. 2011. *Cited in Sec.* 2.1, 2.2.1

[GR05] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution", in *Proceedings of IEEE International Conference on Computer Communications*, Mar. 2005. *Cited in Sec.* 1.2.2, 4.2

[GSD+12] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Wireless video content delivery through coded distributed caching", in *Proceedings of IEEE International Conference on Communications*, pp. 2467–2472, Ottawa, ON, Canada, Jun. 2012. *Cited in Sec.* 4.2

[GVK98] V. Goyal, M. Vetterli, and J. Kovačević, "Multiple description transform coding: robustness to erasures using tight frame expansions", in *Proceedings of IEEE International Symposium on Information Theory*, Cambridge, MA, USA, Aug. 1998. *Cited in Sec.* 2.1

[GVT98] V. Goyal, M. Vetterli, and N. Thao, "Quantized overcomplete expansions in R^N: analysis, synthesis, and algorithms". *IEEE Transactions on Information Theory*, vol. 44 (1), pp. 16–31, Jan. 1998. *Cited in Sec.* 2.1

[HCCF12] H. Huang, S.-H. Chan, G. Cheung, and P. Frossard, "Near-optimal content replication for interactive multi-view video streaming", in *Proceedings of IEEE Packet Video Conference*, pp. 95–100, Munich, Germany, May 2012. *Cited in Sec.* 4.2, 4.4

[HMS+03] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding", in *Proceedings of IEEE International Symposium on Information Theory*, Kanagawa, Japan, Jun. 2003. *Cited in Sec.* (document), 1.1.4

[HO00] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications". *Elsevier Journal on Neural Networks*, vol. 13 (4–5), pp. 411–430, May 2000. *Cited in Sec.* 5.1

[HRM11] S. Huang, A. Ramamoorthy, and M. Medard, "Minimum cost mirror sites using network coding: Replication versus coding at the source nodes". *IEEE Transactions on Information Theory*, vol. 57 (2), pp. 1080–1091, Feb. 2011. *Cited in Sec.* 4.2

[HZC+12] H. Huang, B. Zhang, S. Chan, G. Cheung, and P. Frossard, "Coding and caching co-design for interactive multiview video streaming", in *Proceedings of IEEE International Conference on Computer Communications*, Orlando, FL, USA, Mar. 2012. *Cited in Sec.* 4.2

[IKLAA11] L. Iwaza, M. Kieffer, L. Liberti, and K. Al Agha, "Joint decoding of multiple-description network-coded data", in *Proceedings of IEEE International Symposium on Network Coding*, Beijing, PRC, Jul. 2011. *Cited in Sec.* 2.2

[JSC+05] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction". *IEEE Transactions on Information Theory*, vol. 51 (6), pp. 1973–1982, Jun. 2005. *Cited in Sec.* 1.1.3

[KK08] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding". *IEEE Transactions on Information Theory*, vol. 54 (8), pp. 3579–3591, Aug. 2008. *Cited in Sec.* 1.5

[KLL+97] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web", in *Proceedings of ACM Symposium on Theory of Computing*, pp. 654–663, El Paso, TX, USA, May 1997. *Cited in Sec.* 4.1

[KLWK12] Y.-C. Kao, C.-N. Lee, P.-J. Wu, and H.-H. Kao, "A network coding equivalent content distribution scheme for efficient peer-to-peer interactive VoD streaming". *IEEE Transactions on Parallel and Distributed Systems*, vol. 23 (6), pp. 985–994, Jun. 2012. *Cited in Sec.* 4.2, 4.4

[KM03] R. Koetter and M. Médard, "An algebraic approach to network coding". *IEEE/ACM Transactions on Networking*, vol. 11 (5), pp. 782–795, Oct. 2003. *Cited in Sec.* 1.1.3

[KRH+08] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding". *IEEE/ACM Transactions on Networking*, vol. 16 (3), pp. 497–510, Jun. 2008. *Cited in Sec.* 1.3

[LYC03] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding". *IEEE Transactions on Information Theory*, vol. 49 (2), pp. 371–381, Feb. 2003. *Cited in Sec.* 1.1.3

[MPE11] I. J. MPEG2011/N12036, "Call for proposals on 3D video coding technology", Mar. 2011. *Cited in Sec.* 3.1.2

[MSMW07] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17 (11), pp. 1461–1473, Nov. 2007, Invited Paper. *Cited in Sec.* 3.1.2

[NSV12] S. Nazir, V. Stankovic, and D. Vukobratovic, "Adaptive layered multiple description coding for wireless video with expanding window random linear codes", in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1353–1356, Kyoto, Japan, Mar. 2012. *Cited in Sec.* 2.3

[OWVR97] M. Orchard, Y. Wang, V. Vaishampayan, and A. Reibman, "Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms", in *Proceedings of IEEE Internantional Conference on Image Processing*, Washington, DC, USA, Oct. 1997. *Cited in Sec.* 2.1

[PCDPP11] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Using distributed source coding and depth image based rendering to improve interactive multiview video access", in *Proceedings of IEEE Internantional Conference on Image Processing*, pp. 597–600, Sep. 2011. *Cited in Sec.* 4.1

[PRB+09] P. Parate, L. Ramaswamy, S. Bhandarkar, S. Chattopadhyay, and H. Devulapally, "Efficient dissemination of personalized video content in resource-constrained environments", in *Proceedings of IEEE International Conference on Collaborative Computing*, pp. 1–9, Washington, DC, USA, Nov. 2009. *Cited in Sec.* 4.1, 4.4

[PRV08] S. Plass, G. Richter, and A. H. Vinck, "Coding schemes for crisscross error patterns". *Wireless Personal Communications (Springer)*, vol. 47 (1), pp. 39–49, 2008. *Cited in Sec.* 1.5, 1.5.1, 2

[PS02] V. N. Padmanabhan and K. Sripanidkulchai, "The case for cooperative networking", in *Proceedings of International Workshop on Peer-to-Peer Systems*, Cambridge, MA, USA, Mar. 2002. *Cited in Sec.* 4.1

[Reu80] D. O. Reudink, "The channel splitting problem with interpolative coders", Tech. Rep., Bell Labs, Oct. 1980, tM80-134-1. *Cited in Sec.* 2.1

[RF07] I. Radulovic and P. Frossard, "Multiple description image coding with redundant expansions and optimal quantization", in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Crete, Greece, Oct. 2007. *Cited in Sec.* 2.1

[RW10] A. Ramasubramonian and J. Woods, "Multiple description coding and practical network coding for video multicast". *IEEE Signal Processing Letters*, vol. 17 (3), pp. 265–268, Mar. 2010. *Cited in Sec.* 2.2

[SBB+12] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Muller, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D video coding using advanced prediction, depth modeling, and encoder control methods", in *Proceedings of Picture Coding Symposium*, May 2012. *Cited in Sec.* (document), 4.1, 4.5

[Sch80] J. Schwartz, "Fast probabilistic algorithm for verification of polynomial identities". *Journal of the ACM*, vol. 27 (1), pp. 701–717, 1980. *Cited in Sec.* 1.1.3

[SM09] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks". *IEEE Journal on Selected Areas in Communications*, vol. 27 (5), pp. 713–728, Jun. 2009. *Cited in Sec.* 1.4

[SO10] G. Sullivan and J. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)", in *Proceedings of SPIE Conference on Applications of Digital Image Processing*, San Diego, CA, USA, Aug. 2010. *Cited in Sec.* 3.1.2

[Süh11] K. Sühring, "JM reference software release 17.0", Source Code, Jan. 2011. *Cited in Sec.* 2.3.1

[TCF11] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized distributed video delivery with randomized network coding". *IEEE Transactions on Multimedia*, vol. 13 (4), pp. 776–787, Aug. 2011. *Cited in Sec.* 1.4

[TPPP07] C. Tillier, C. T. Petrişor, and B. Pesquet-Popescu, "A motion-compensated overcomplete temporal decomposition for multiple description scalable video coding". *EURASIP Journal on Image and Video Processing*, vol. 1, pp. 1–12, 2007. *Cited in Sec.* 2.1

[TTFY11] M. Tanimoto, M. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV". *IEEE Signal Processing Magazine*, vol. 28 (1), pp. 67–76, Jan. 2011. *Cited in Sec.* 3.1, 4.1

[Vai93] V. Vaishampayan, "Design of multiple description scalar quantizers". *IEEE Transactions on Information Theory*, vol. 39 (3), pp. 821–834, May 1993. *Cited in Sec.* 2.1

[VD94] V. Vaishampayan and J. Domaszewicz, "Design of entropy constrained multiple description scalar quantizers". *IEEE Transactions on Information Theory*, vol. 40 (1), pp. 245–250, Jan. 1994. *Cited in Sec.* 2.1

[VS10] D. Vukobratović and V. Stanković, "Unequal error protection random linear coding for multimedia communications", in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Saint-Malo, France, Oct. 2010. *Cited in Sec.* (document), 2.3, 3.2

[VS12] ———, "Unequal error protection random linear coding strategies for erasure channels". *IEEE Transactions on Communications*, vol. 60 (5), pp. 1243–1252, May 2012. *Cited in Sec.* (document), 2.3

[VWS11] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard". *Proceedings of the IEEE*, vol. 99 (4), pp. 626–642, Apr. 2011, Invited Paper. *Cited in Sec.* 3.1.2, 4.1, 4.3.1

[WFLB05] J. Widmer, C. Fragouli, and J. Le Boudec, "Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding", in *Proceedings of IEEE Workshop on Network Coding, Theory and Applications*, Riva del Garda, Italy, Apr. 2005. *Cited in Sec.* 1.3

[WL07a] M. Wang and B. Li, "Lava: a reality check of network coding in peer-to-peer live streaming", in *Proceedings of IEEE International Conference on Computer Communications*, Anchorage, AK, USA, May 2007. *Cited in Sec.* 1.2.2

[WL07b] ———, "R^2: random push with random network coding in live peer-to-peer streaming". *IEEE Journal on Selected Areas in Communications*, vol. 25 (9), pp. 1655–1666, Dec. 2007. *Cited in Sec.* 1.2.2

[WOR97] Y. Wang, M. Orchard, and A. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients", in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Princeton, NJ, USA, Jun. 1997. *Cited in Sec.* 2.1

[WOVR01] Y. Wang, M. Orchard, V. Vaishampayan, and A. Reibman, "Multiple description coding using pairwise correlating transforms". *IEEE Transactions on Image Processing*, vol. 10 (3), pp. 351–366, Mar. 2001. *Cited in Sec.* 2.1

[WRL05] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery". *Proceedings of the IEEE*, vol. 93 (1), pp. 57–70, Jan. 2005, Invited Paper. *Cited in Sec.* 2.1

[Wu10] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage". *IEEE Journal on Selected Areas in Communications*, vol. 28 (2), pp. 277–288, 2010. *Cited in Sec.* 1.2.1

[WZLJ10] H. Wang, Y. Zhang, P. Li, and Z. Jiang, "The benefits of network coding in distributed caching in large-scale P2P-VoD systems", in *Proceedings of IEEE Global Telecommunications Conference*, pp. 1–6, Miami, FL, USA, Dec. 2010. *Cited in Sec.* 4.2

[Yer07] A. Yeredor, "ICA in Boolean XOR mixtures", in *Proceedings of Springer-Verlag Internantional Conference on Independent Component Analysis*, pp. 827–835, 2007. *Cited in Sec.* 5.1, 5.2.2, 5.3.1

[Yer11] ———, "Independent component analysis over galois fields of prime order". *IEEE Transactions on Information Theory*, vol. 57 (8), pp. 5342–5359, Aug. 2011. *Cited in Sec.* 5.1

CODAGE RÉSEAU POUR DES APPLICATIONS MULTIMÉDIAS AVANCÉES

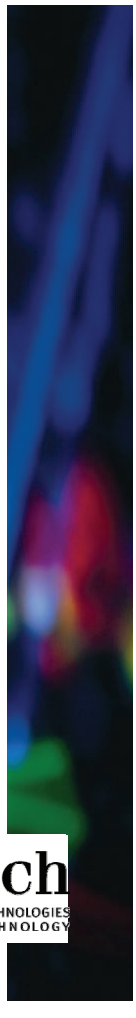
Irina-Delia NEMOIANU

RESUMÉ : Le codage réseau permet une utilisation efficace du réseau. Il maximise le débit dans un réseau multi-saut en multicast et réduit le retard. Dans cette thèse, nous concentrons notre attention sur l'intégration du codage réseau aux applications multimédias, et en particulier aux systèmes qui fournissent un service vidéo amélioré. Nos contributions concernent plusieurs scénarios : un cadre de fonctions efficace pour la transmission de flux en directe qui utilise à la fois le codage réseau et le codage par description multiple, une nouvelle stratégie de transmission pour les réseaux sans fil avec perte qui garantit un compromis entre la résilience vis-à-vis des perte et la reduction du retard sur la base d'une optimisation débit-distorsion de l'ordonnancement des images vidéo, que nous avons également étendu au cas du streaming multi-vue interactive, un système replication sociale distribuée qui, en utilisant le réseau codage et la connaissance des préférences des utilisateurs en termes de vue, est en mesure de sélectionner un schéma de réplication capable de fournir une vidéo de haute qualité en accédant seulement aux autres membres du groupe social, sans encourir le coût d'accès associé à une connexion à un serveur central et sans échanger des larges tables de métadonnées pour tenir trace des éléments répliqués, et, finalement, une étude sur l'utilisation de techniques de séparation aveugle de source –pour réduire l'overhead encouru par les schémas de codage réseau– basé sur des techniques de détection d'erreur telles que le codage de parité et la génération de message digest.

MOTS-CLEFS : Codage réseau, codage par description multiple, vidéo multi-vue, réseaux sans fil, séparation aveugle de source.

ABSTRACT : Network coding is a paradigm that allows an efficient use of networks. It maximizes the throughput in a multi-hop multicast communication and reduces the delay. In this thesis, we focus on the integration of network coding to multimedia applications, and in particular to systems that provide enhanced video services in terms of distortion and delay perceived by the users. Our contributions concern several instances : an efficient framework for transmission of a live stream making joint use of network coding and multiple description coding ; a novel transmission strategy for lossy wireless networks that guarantees a trade-off between loss resilience and short delay based on a rate-distortion optimized scheduling of the video frames, that we also extended to the case of interactive multi-view streaming ; a distributed social caching system that, using network coding in conjunction with the knowledge of the users' preferences in terms of views, is able to select a replication scheme that provides a high video quality by accessing only other members of the social group without the costs associated with connecting to a central server and exchanging large tables of metadata to keep track of the replicated parts ; and, finally, a study on using blind source separation techniques –to reduce the overhead incurred by network coding schemes– based on error-detecting techniques such as parity coding and message digest generation.

KEY-WORDS : Network coding, multiple description coding, multi-view video, wireless networks, blind source separation.



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY